



OKI.565

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of

Teruaki Uehara

Group Art Unit: 2193

Serial No.: 10/642,735

Examiner: C. Ngo

Filed: August 19, 2003

Confir. No.: 3778

For: ARITHMETIC UNIT AND METHOD FOR DATA STORAGE AND READING

CLAIM OF PRIORITY

U.S. Patent and Trademark Office
Customer Window, **Mail Stop Amendment**
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Date: November 22, 2006

Sir:

Applicant, in the above-identified application, hereby claims the priority date
under the International Convention of the following Japanese application:

Appln. No. 2002-241371

filed August 22, 2002

as acknowledged in the Declaration of the subject application.

A certified copy of said application is being submitted herewith.

Respectfully submitted,

VOLENTINE FRANCOS & WHITT, P.L.L.C.

Andrew J. Telesz, Jr.
Registration No. 33,581

One Freedom Square
11951 Freedom Drive, Suite 1260
Reston, Virginia 20190
Tel. (571) 283-0720
Fax. (571) 283-0740

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出 願 年 月 日
Date of Application:

2002年 8月22日

出 願 番 号
Application Number:

特願2002-241371

ST.10/C]:

[JP2002-241371]

出 願 人
Applicant(s):

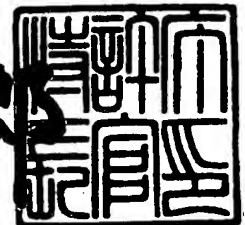
沖電気工業株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2003年 1月14日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



【書類名】 特許願

【整理番号】 OH003719

【あて先】 特許庁長官殿

【国際特許分類】 G06F 5/00
G05B 19/00

【発明者】

【住所又は居所】 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社
社内

【氏名】 上原 輝昭

【特許出願人】

【識別番号】 000000295

【氏名又は名称】 沖電気工業株式会社

【代理人】

【識別番号】 100085419

【弁理士】

【氏名又は名称】 大垣 孝

【手数料の表示】

【予納台帳番号】 012715

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9001068

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 演算装置及びデータの格納方法及びデータの読出方法

【特許請求の範囲】

【請求項 1】 メモリから読み出したデータに対して所定の演算を行う演算論理ユニットと、

前記メモリから読み出したデータを一時格納するレジスタと、

前記演算論理ユニットまたは前記レジスタを選択し、前記メモリから読み出したデータの一部を、選択した前記演算論理ユニットまたは前記レジスタから出力される出力データに置き換えることが可能な混合回路と、
を有することを特徴とする演算装置。

【請求項 2】 桁上げ信号が任意のビット位置での桁上げを指示する場合に、上位桁へのキャリー伝搬を禁止することが可能な演算論理ユニットと、

前記演算論理ユニットでの演算前に、前記演算論理ユニットで使用されるデータを格納可能なレジスタと、

前記演算論理ユニットまたは前記レジスタを選択し、メモリから読み出したデータの一部を、選択した前記演算論理ユニットまたは前記レジスタから出力される出力データに置き換えることが可能な混合回路と、
を有することを特徴とする演算装置。

【請求項 3】 複数のデータを連続して格納するメモリと、

所定の組合せによるデータのビット幅が 2 の n 乗ビット未満である場合に、後続の所定の組合せによるデータの先頭が 0 ビット目または 8 ビット目になるように、0 ビット目または 8 ビット目から不足するビット幅分だけ未使用データを所定の組合せによるデータの後に挿入する混合回路と、
を有することを特徴とする演算装置。

【請求項 4】 前記混合回路は、前記メモリから読み出したデータの一部を、前記演算論理ユニットが演算した演算結果に置き換える場合に、一演算処理毎に、置き換えるデータの位置を所定ビットづつシフトして置き換えることを特徴とする請求項 1 ～ 3 のいずれか 1 項に記載の演算装置。

【請求項 5】 前記メモリは 2 つのメモリブロックを備えることを特徴とす

る請求項 1 ～ 4 のいずれか 1 項に記載の演算装置。

【請求項 6】 所定の演算を行う演算論理ユニットとデータを格納するメモリとを有する演算回路が、複数のデータを連続して前記メモリに格納する場合に、所定の組合せによるデータのビット幅が 2 の n 乗ビット未満であるときに、後続の所定の組合せによるデータの先頭が 0 ビット目または 8 ビット目になるように、0 ビット目または 8 ビット目から不足するビット幅分だけ未使用データを所定の組合せによるデータの後に挿入することを特徴とするデータの格納方法。

【請求項 7】 前記演算回路が、前記メモリから読み出したデータの一部を、前記演算論理ユニットが演算した演算結果に置き換える場合に、一演算処理毎に、置き換えるデータの位置を所定ビットづつシフトして置き換えることを特徴とする請求項 6 に記載のデータの格納方法。

【請求項 8】 所定の演算を行う演算論理ユニットとデータを格納するメモリとを有する演算回路が、第 1 及び第 2 の 2 つのメモリブロックを備えるメモリから 2 の n 乗ビットづつデータを読み出す工程と、読み出したデータの中から演算処理に供さない部分と演算処理に供する部分とに分離する工程とを有し、

前記読み出す工程では、同一段の第 1 及び第 2 のメモリブロックから読み出す場合と、第 2 のメモリブロック及び次段の第 1 のメモリブロックから読み出す場合とを所定の順番で行い、

前記分離する工程では、一演算処理毎に、前記メモリから読み出したデータから演算処理に供さない部分の位置を所定ビットづつシフトして演算処理に供する部分と演算処理に供さない部分とに分離することを特徴とするデータの読出方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、1 ワードのビット幅が標準的な 2 の n 乗ビットでないデータの演算処理を行う演算処理装置に関する。

【0002】

【従来の技術】

画像や音声等を入力するまたは加工する装置の中には、画像や音声の品位を向上させるために、あるいは画像や音声に付加的な情報を添付するために、1ワードのビット幅が標準的な2のn乗ビットでないデータを用いるものがある。このような装置として、例えば、いわゆる第3世代携帯電話や複数階調の画像データを発生する情報処理装置等がある。このような装置は、デジタルシグナルプロセッサ（以下、DSPという）やその他の演算装置を搭載しており、これにより1ワードのビット幅を標準的な2のn乗ビットに変換して様々な演算処理を行う。

【0003】

以下に、第3世代携帯電話に搭載されるDSPを例にして、従来の演算装置を説明する。

【0004】

第3世代携帯電話は、常に良好な通信が行なえるように、DSPを用いて広い周波数帯域の中からいくつかの特定帯域の信号を抽出する。そして、その中から特に受信感度の高い周波数帯域を選択して通信を行なう。なお、このとき、DSPは、通常、後述のデジタルマッチドフィルタ（以下、DMFという）アルゴリズムと称される手法を用いて特定帯域の信号のパス強度を強調し、これにより特定帯域の信号を抽出する。

【0005】

図19は従来の演算装置の構成を示す図、図20及び図21は2つのメモリブロックに格納されるデータの配置を示す図、図22及び図23はDMFアルゴリズムの概要を示す図、図24～図27はDMFアルゴリズムの算術式を示す図表である。なお、ここでは、各ブロック間でのデータの伝送処理や演算手段での演算処理は32ビット単位で行われるものとして説明する。

【0006】

図19中、シフター11は、演算処理を容易化させるために、後述のアクキュレータ15やメモリ17から伝送されるデータを所定ビットシフトして位相の調整を行う。演算論理ユニット（以下、ALUという）13は、シフター11から伝送されるデータや後述のアクキュレータ15から伝送されるデータに基づいて、後述のDMFアルゴリズムによる演算を実行する。アクキュレータ15はAL

U 1 3 による演算結果を格納するレジスタであり、 $A c c_0$ と $A c c_1$ の2つのブロックからなる。なお、アキュムレータ 1 5 に格納された演算結果は、第 1 のルート 2 3 に沿ってメモリ 1 7 に出力される。また、図示しない制御部の制御に基づいて、所定の演算処理時に第 2 のルート 2 5 に沿って A L U 1 3 に伝送され、そこで後続の演算処理に供される。

【 0 0 0 7 】

メモリ 1 7 は、所定ビット（図 1 9 に示す従来例では 1 6 ビット）幅の第 1 のメモリブロック 1 9 と第 2 のメモリブロック 2 1 を備え、アキュムレータ 1 5 から伝送される A L U 1 3 による演算結果を格納する。

【 0 0 0 8 】

なお、図 2 0 にメモリ 1 7 に格納されるデータの大まかな配置を、更に図 2 1 にメモリ 1 7 に格納されるデータの詳細な配置を示す。メモリ 1 7 は、I パートデータと R パートデータの組合せによる演算結果のうち、I パートデータを第 1 のメモリブロック 1 9 に、R パートデータを第 2 のメモリブロック 2 1 に格納する。そして、メモリ 1 7 は、演算結果の各ワードが所定ビット（1 6 ビット）幅未満である場合に、不足するビット幅分だけ未使用データを格納する。例えば、図 2 0 及び図 2 1 に示す例では、I パートデータと R パートデータがそれぞれ 1 0 ビット幅であるので、メモリ 1 7 は 6 ビット幅の未使用データを第 1 のメモリブロック 1 9 と第 2 のメモリブロック 2 1 のそれぞれに格納している。なお、図 2 1 中、「 $[-I n-]$ 」は I パートデータを格納している領域、「 $[-R n-]$ 」は R パートデータを格納している領域、「 $*$ 」は未使用データを格納している領域を示している。

【 0 0 0 9 】

なお、1 6 ビット幅のメモリ 1 7 に 3 2 ビット幅の A L U 1 3 を接続するのは、乗算演算により演算幅が増大した場合に対応するためであるが、メモリ 1 7 は 1 6 ビット幅に限らず、4 や 3 2 等、2 の n 乗のビット幅にすることも可能である。ただし、現時点では、1 6 ビット幅のメモリが広く普及している上、コストも安いので、D S P のコストを低減させるには 1 6 ビット幅のものが望ましい。

【 0 0 1 0 】

以下に、DMFアルゴリズムについて説明する。図22及び図23はDMFアルゴリズムの概要を示す図、図24～図27はDMFアルゴリズムの算術式を示す図表である。

【0011】

図22中、「+」は加算演算を示し、「*」は乗算演算を示す。ただし、乗算演算*は、DMFアルゴリズムにおける乗数が1または-1であるため、実際は符号が変わらないか反転することになるだけである。なお、加算回路や乗算回路の構成については特に問わないので、ここでは説明を省略する。

【0012】

$D_1 \cdots D_8$ はデータの遅延値を示し、 D_1 の出力データは128サイクル前の入力データ、 D_2 の出力データは64サイクル前の入力データ、 D_3 の出力データは16サイクル前の入力データ、 D_4 の出力データは3.2サイクル前の入力データ、 D_5 の出力データは8サイクル前の入力データ、 D_6 の出力データは1サイクル前の入力データ、 D_7 の出力データは4サイクル前の入力データ、 D_8 の出力データは2サイクル前の入力データとなる。 $W_1 \cdots W_8$ は遅延値Dに乘算される乗算係数を示し、 W_1 は1、 W_2 は-1、 W_3 は1、 W_4 は1、 W_5 は1、 W_6 は1、 W_7 は1、 W_8 は1となる。これら遅延値 $D_1 \cdots D_8$ や乗算係数 $W_1 \cdots W_8$ の値は、演算結果が特定の遅延値の信号（すなわち、特定帯域の信号）のパス強度を強調した値となるように、固定的に設定されている。ALU13はDMFアルゴリズムを用いた演算を行うことにより特定帯域の信号を強調することができ、これによって特定帯域の信号を抽出可能となる。

【0013】

図23は図22における各演算過程の入出力を示し、図24～図27は図23における各入出力の関係を示している。図23に示すように、各入出力 $A_1 \sim A_8$ 、 $B_1 \sim B_8$ 、 $C_1 \sim C_8$ 、 $C_1' \sim C_3'$ 、 C_5' 、 C_7' は、入力 A_0 を起点としている。各入出力の関係を表す算術式を図24に、また、 $B_1 \sim B_8$ の値を図25に、 $C_1 \sim C_8$ の値を図26に、 $C_1' \sim C_3'$ 、 C_5' 及び C_7' の値を図27に示す。

【0014】

なお、遅延値 $D_1 \sim D_8$ の各出力 $A_1 \sim A_8$ は、遅延値 $D_1 \sim D_8$ の値がそれぞれ 1 2 8, 6 4, 1 6, 3 2, 8, 1, 4, 2 サイクルであるので、 A_0 の値を起点にしてそれぞれ 1 2 8, $1\ 2\ 8 + 6\ 4 = 1\ 9\ 2$, $1\ 9\ 2 + 1\ 6 = 2\ 0\ 8$, $2\ 0\ 8 + 3\ 2 = 2\ 4\ 0$, $2\ 4\ 0 + 8 = 2\ 4\ 8$, $2\ 4\ 8 + 1 = 2\ 4\ 9$, $2\ 4\ 9 + 4 = 2\ 5\ 3$, $2\ 5\ 3 + 2 = 2\ 5\ 5$ サイクル遅延した値となる。例えば入力 A_0 の値を x_0 とする場合に、 D_1 の出力 A_1 の値は x_{128} 、 D_2 の出力 A_2 の値は x_{192} 、 D_3 の出力 A_3 の値は x_{208} 、 D_4 の出力 A_4 の値は x_{240} 、 D_5 の出力 A_5 の値は x_{248} 、 D_6 の出力 A_6 の値は x_{249} 、 D_7 の出力 A_7 の値は x_{253} 、 D_8 の出力 A_8 の値は x_{255} となる。

【 0 0 1 5 】

DMF アルゴリズムによる演算結果は、DMF アルゴリズムの上の段による演算結果（例えば $C_1 \sim C_8$ の値）はアキュムレータ 1 5 の $A c c_1$ に格納され、下の段による演算結果（例えば $C_1' \sim C_3'$ 及び C_5' 及び C_7' の値）はアキュムレータ 1 5 の $A c c_0$ に格納される。

【 0 0 1 6 】

【発明が解決しようとする課題】

従来の DSP は、一度に 1 6 ビットのデータをメモリ 1 7 から ALU 1 3 に出力する。しかしながら、演算処理に用いるデータは、そのうちの 1 0 ビット分だけである。そのため、従来の演算装置は、一度に 6 ビットのデータを無駄に ALU 1 3 に出力していた。

【 0 0 1 7 】

また ALU 1 3 は 3 2 ビット幅の演算器（図示せず）を内蔵するが、ALU 1 3 は、そのうちの 1 0 ビット幅分しか使用していない。そのため、従来の演算装置は、2 2 ビット幅分の演算器を無駄にしていた。

【 0 0 1 8 】

このように、従来の演算装置は、1 ワードが標準的なビット幅でないデータを用いて演算処理する場合に、I パートデータと R パートデータの間に未使用部を配置しているため ALU 1 3 に内蔵される演算器やメモリ 1 7 の使用に無駄が発生し、特に DMF アルゴリズムによる演算処理を行なう際に、演算能力やメモリ

を有効に活用できないという問題点があった。

【 0 0 1 9 】

【課題を解決するための手段】

上記課題を解決するために本発明に係る演算装置は、メモリから読み出したデータに対して所定の演算を行う演算論理ユニットと、前記メモリから読み出したデータを一時格納するレジスタと、前記演算論理ユニットまたは前記レジスタを選択し、前記メモリから読み出したデータの一部を、選択した前記演算論理ユニットまたは前記レジスタから出力される出力データに置き換えることが可能な混合回路と、を有することを特徴とする。

【 0 0 2 0 】

また別の本発明に係る演算装置は、桁上げ信号が任意のビット位置での桁上げを指示する場合に、上位桁へのキャリー伝搬を禁止することが可能な演算論理ユニットと、前記演算論理ユニットでの演算前に、前記演算論理ユニットで使用されるデータを格納可能なレジスタと、前記演算論理ユニットまたは前記レジスタを選択し、メモリから読み出したデータの一部を、選択した前記演算論理ユニットまたは前記レジスタから出力される出力データに置き換えることが可能な混合回路と、を有することを特徴とする。

【 0 0 2 1 】

さらに別の本発明に係る演算装置は、複数のデータを連続して格納するメモリと、所定の組合せによるデータのビット幅が2のn乗ビット未満である場合に、後続の所定の組合せによるデータの先頭が0ビット目または8ビット目になるように、0ビット目または8ビット目から不足するビット幅分だけ未使用データを所定の組合せによるデータの後に挿入する混合回路と、を有することを特徴とする。

【 0 0 2 2 】

【発明の実施の形態】

本実施の形態は、16ビット幅のレジスタと混合回路を設け、制御部に特殊な制御をさせることにより、使用メモリ量（特に、DMFアルゴリズムに基づく演算処理時における使用メモリ量）を従来の2/3に低減することを目的とする。

【 0 0 2 3 】

図 1 は本発明に係る演算装置の構成を示す図である。

【 0 0 2 4 】

本発明に係る演算装置は、1ワードが標準的な2のn乗のビット幅でないデータの演算を、高速に少ないメモリ量で行えるように構成されている。

【 0 0 2 5 】

以下に、第3世代携帯電話に搭載されるDSPを例にして、本発明の実施の形態を説明する。なお、各図は本発明を理解できる程度に概略的に示してあるにすぎない。また、各図において、共通する要素については、同一の符号を付し、説明を省略する。

【 0 0 2 6 】

本実施の形態に係るDSPは、図1に示すように、レジスタ27と混合回路29とを備えている。レジスタ27は、メモリ17から出力されるデータを一時格納するために設けられている。混合回路29は、メモリ17から出力されるデータ（すなわち、レジスタ27に一時格納されたデータ）に対し、その一部をALU13'から出力されるデータに置き換えるために設けられている。なお、レジスタ27と混合回路29も、他の要素と同様に、図示しない制御部によって制御されている。また、本実施の形態に係るALU13'は、後述するように、分割信号Kが入力されることにより任意の位置でキャリー信号を分割する機能が付加されている。

【 0 0 2 7 】

ところで、本実施の形態は、メモリ17として、既に大量に普及している汎用的なメモリを利用可能とすることが意図されている。そこで、本実施の形態では、メモリ17に格納するデータを図2～図12に示すように配列するとともに、ALU13'が演算し易くなるように、図示しない制御部によってメモリ17に対して特殊な制御を行っている。

【 0 0 2 8 】

図2～図12は、2つのメモリブロックに格納されるデータの配置を示す図であり、図2にはメモリ17に格納されるデータの大まかな配置が、また図3～図

12にはメモリ17に格納されるデータの詳細な配置が示されている。なお、図3～図12中、「 $[-I_n-]$ 」はIパートデータを格納している領域、「 $[-R_n-]$ 」はRパートデータを格納している領域、「 $*$ 」は未使用データを格納している領域を示している。

【0029】

図2～図12に示すように、メモリ17は、10ビット幅のIパートデータと10ビット幅のRパートデータの組合せによるデータに4ビット幅の未使用データを付加して24ビット幅のデータとし、これらを各16ビット幅の第1及び第2のメモリブロック19、21の中に連続して格納する。本実施の形態は、データをこのように格納することにより、メモリ17の未使用領域を従来の $12/32 = 37.5\%$ 分から $4/32 = 12.5\%$ 分に低減している。

【0030】

なお、未使用データのビット幅は、後述の理由により、後続のデータの先頭が0ビット目または8ビット目から始まるようにするのが望ましい。例えば、図2に示す例では、Iパートデータのビット幅が10ビット、Rパートデータのビット幅が10ビットであるので、未使用データのビット幅は4ビットになっている。仮に、IパートデータやRパートデータのビット幅が変われば、これに応じて後続のデータの先頭が0ビット目または8ビット目から始まるように、未使用データのビット幅も変えるのが望ましい。

【0031】

本実施の形態では、図示しない制御部は、メモリ17に対して以下のような制御を行う。

【0032】

すなわち、例えば、メモリ17には図3～図12に示すようなデータが格納されているものとする。

【0033】

図示しない制御部は、図22及び図23に示すDMFアルゴリズムに基づいて入力 A_0 をALU13'に供給する。ここでは、入力 A_0 は、I0パートデータとR0パートデータとする。I0パートデータとR0パートデータは、メモリ17

のアドレス0と1に格納されている。そこで、図示しない制御部は、メモリ17のアドレス0と1に格納されていたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をアキュムレータ（以下、Accという）15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に、後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス0と1に格納させる。

【0034】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_1 の出力 A_1 をALU13'に供給する。ここでは、出力 A_1 は、遅延値 D_1 の値が128サイクルであるので $0 + 128 = 128$ サイクル遅延して格納されたデータとなる。すなわち、I128パートデータとR128パートデータとなる。I128パートデータとR128パートデータは、メモリ17のアドレス192と193に格納されている。そこで、図示しない制御部は、メモリ17のアドレス192と193に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス192と193に格納させる。

【0035】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_2 の出力 A_2 をALU13'に供給する。ここでは、出力 A_2 は、遅延値 D_2 の値が64サイクルであるので $128 + 64 = 192$ サイクル遅延して格納されたデータとな

る。すなわち、I192パートデータとR192パートデータとなる。I192パートデータとR192パートデータは、メモリ17のアドレス288と289に格納されている。そこで、図示しない制御部は、メモリ17のアドレス288と289に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス288と289に格納させる。

【0036】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_3 の出力 A_3 をALU13'に供給する。ここでは、出力 A_3 は、遅延値 D_3 の値が16サイクルであるので $192 + 16 = 208$ サイクル遅延して格納されたデータとなる。すなわち、I208パートデータとR208パートデータとなる。I208パートデータとR208パートデータは、メモリ17のアドレス312と313に格納されている。そこで、図示しない制御部は、メモリ17のアドレス312と313に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス312と313に格納させる。

【0037】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_4 の出力 A_4 をALU13'に供給する。ここでは、出力 A_4 は、遅延値 D_4 の値が32サ

イクルであるので $208 + 32 = 240$ サイクル遅延して格納されたデータとなる。すなわち、I 2 4 0 パートデータと R 2 4 0 パートデータとなる。I 2 4 0 パートデータと R 2 4 0 パートデータは、メモリ 1 7 のアドレス 3 6 0 と 3 6 1 に格納されている。そこで、図示しない制御部は、メモリ 1 7 のアドレス 3 6 0 と 3 6 1 に格納されたデータを、シフター 1 1 を介して A L U 1 3' に供給するとともに、レジスタ 2 7 にも供給してそこに一時格納させる。次に、図示しない制御部は、A L U 1 3' に一演算処理を実行させ、その演算結果を A c c 1 5 に一時格納させる。次に、レジスタ 2 7 に一時格納させたデータを混合回路 2 9 に出力させるとともに、A c c 1 5 に一時格納された演算結果を混合回路 2 9 に出力させる。次に、混合回路 2 9 に後述する制御によって両者を混合させ、混合結果をメモリ 1 7 に出力させる。次に、メモリ 1 7 に、混合結果を元のアドレスであるアドレス 3 6 0 と 3 6 1 に格納させる。

【 0 0 3 8 】

また、図示しない制御部は、DMF アルゴリズムに基づいて遅延値 D_5 の出力 A_5 を A L U 1 3' に供給する。ここでは、出力 A_5 は、遅延値 D_5 の値が 8 サイクルであるので $240 + 8 = 248$ サイクル遅延して格納されたデータとなる。すなわち、I 2 4 8 パートデータと R 2 4 8 パートデータとなる。I 2 4 8 パートデータと R 2 4 8 パートデータは、メモリ 1 7 のアドレス 3 7 2 と 3 7 3 に格納されている。そこで、図示しない制御部は、メモリ 1 7 のアドレス 3 7 2 と 3 7 3 に格納されたデータを、シフター 1 1 を介して A L U 1 3' に供給するとともに、レジスタ 2 7 にも供給してそこに一時格納させる。次に、図示しない制御部は、A L U 1 3' に一演算処理を実行させ、その演算結果を A c c 1 5 に一時格納させる。次に、レジスタ 2 7 に一時格納させたデータを混合回路 2 9 に出力させるとともに、A c c 1 5 に一時格納された演算結果を混合回路 2 9 に出力させる。次に、混合回路 2 9 に後述する制御によって両者を混合させ、混合結果をメモリ 1 7 に出力させる。次に、メモリ 1 7 に、混合結果を元のアドレスであるアドレス 3 7 2 と 3 7 3 に格納させる。

【 0 0 3 9 】

また、図示しない制御部は、DMF アルゴリズムに基づいて遅延値 D_6 の出力

A_6 をALU13'に供給する。ここでは、出力 A_6 は、遅延値 D_6 の値が1サイクルであるので $248 + 1 = 249$ サイクル遅延して格納されたデータとなる。すなわち、I249パートデータとR249パートデータとなる。I249パートデータとR249パートデータは、メモリ17のアドレス373と374に格納されている。そこで、図示しない制御部は、メモリ17のアドレス373と374に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス373と374に格納させる。

【0040】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_7 の出力 A_7 をALU13'に供給する。ここでは、出力 A_7 は、遅延値 D_7 の値が4サイクルであるので $249 + 4 = 253$ サイクル遅延して格納されたデータとなる。すなわち、I253パートデータとR253パートデータとなる。I253パートデータとR253パートデータは、メモリ17のアドレス379と380に格納されている。そこで、図示しない制御部は、メモリ17のアドレス379と380に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス379と380に格納させる。

【0041】

また、図示しない制御部は、DMFアルゴリズムに基づいて遅延値 D_8 の出力 A_8 をALU13'に供給する。ここでは、出力 A_8 は、遅延値 D_8 の値が2サイクルであるので $253 + 2 = 255$ サイクル遅延して格納されたデータとなる。すなわち、I255パートデータとR255パートデータとなる。I255パートデータとR255パートデータは、メモリ17のアドレス382と383に格納されている。そこで、図示しない制御部は、メモリ17のアドレス382と383に格納されたデータを、シフター11を介してALU13'に供給するとともに、レジスタ27にも供給してそこに一時格納させる。次に、図示しない制御部は、ALU13'に一演算処理を実行させ、その演算結果をAcc15に一時格納させる。次に、レジスタ27に一時格納させたデータを混合回路29に出力させるとともに、Acc15に一時格納された演算結果を混合回路29に出力させる。次に、混合回路29に後述する制御によって両者を混合させ、混合結果をメモリ17に出力させる。次に、メモリ17に、混合結果を元のアドレスであるアドレス382と383に格納させる。

【0042】

この後、アドレス383のI255パートデータとR255パートデータには、次の遅延値 D_1 の入力 A_0 が配置される。そして、次の演算処理時において、図示しない制御部は、アドレス383を新たなアドレス0と見なして上述した動作と同様の動作を実行する。

【0043】

なお、上述した動作において、データが遷移するパターンは3通りあり、それぞれを図13(a)～(c)に示す。なお、図13は2つのメモリブロックに格納されるデータの遷移過程を示す図である。図13中、無地の領域はデータが更新される部分を示しており、斜線を付した領域はデータが更新されない部分を示している。

【0044】

図13(a)～(c)に示す各パターンの遷移過程において、DSPの各構成は以下のように動作する。

【0045】

図 1 3 (a) に示すパターン 1 において、まず図示しない制御部は、第 1 のメモリブロック 1 9 からアドレス n に格納されたデータを読み出すとともに、第 2 のメモリブロック 2 1 からアドレス $n + 1$ に格納されたデータを読み出し、シフター 1 1 とレジスタ 2 7 に出力する。なお、アドレス n に格納されたデータとは I 0 パートデータと R 0 パートデータの一部である。またアドレス $n + 1$ に格納されたデータとは R 0 パートデータの一部と未使用データと I 1 パートデータである。

【 0 0 4 6 】

シフター 1 1 は、アキュムレータ 1 5 やメモリ 1 7 から伝送されるデータを所定ビットシフトして位相の調整を行う。そして、その結果を A L U 1 3 ' に出力する。

【 0 0 4 7 】

図 1 4 は本実施の形態に係る A L U 1 3 ' の内部構成を示す図である。図 1 4 中、A と B はシフター 1 1 を介して第 1 及び第 2 のメモリブロック 1 9 、 2 1 から伝送されるデータ、C はキャリー信号、K は分割信号、X は出力信号、F A は加算演算回路である。

【 0 0 4 8 】

A L U 1 3 ' は、シフター 1 1 から伝送されるデータに対し、図示しない制御部から伝送される分割信号 K に基づいて、I 0 パートデータと R 0 パートデータの組合せによるデータ（以下、演算に供するデータという）と、それ以外のデータ（以下、演算に供しないデータという）とに分割し、演算に供するデータを取得する。なお、分割は分割信号 K が 0 のときに行われる。また演算に供するデータは図 1 4 に示す出力信号 X に相当する。

【 0 0 4 9 】

次に、A L U 1 3 ' は、アキュムレータ 1 5 から、前回の、現在の演算処理のサイクルと同じサイクルにおける演算結果（以下、前回サイクルの演算結果という）を取得する。なお、前回サイクルの演算結果とは、I パートデータと R パートデータの組合せによるデータである。そして、A L U 1 3 ' は、シフター 1 1 から取得した演算に供するデータとアキュムレータ 1 5 から取得した前回サイク

ルの演算結果を用いて、上述のDMFアルゴリズムによる演算を実行する。図13(a)では、このときの演算結果を、I0'パートデータとR0'パートデータの組合せによるデータとして示している。

【0050】

ALU13'は、演算結果をアキュムレータ15に出力する。アキュムレータ15はALU13'による演算結果を格納するとともに、演算結果を第1のルート23に沿って混合回路29に出力する。また、図示しない制御部の制御に基づいて所定のタイミングで演算結果を第2のルート25に沿ってALU13'に出力する。

【0051】

他方、レジスタ27は、図示しない制御部の制御に基づいて、所定のタイミングでメモリ17から伝送されたデータを第3のルート23'に沿って混合回路29に出力する。

【0052】

混合回路29は、図示しない制御部の制御に基づいて、第3のルート23'に沿ってレジスタ27から伝送されるデータを8ビット毎に分割し、その中の演算に供する部分（すなわち、第1のメモリブロック19に格納されていた下位8ビットと上位8ビットのデータと、第2のメモリブロック21に格納されていた下位8ビットのデータ）を、第1のルート23に沿ってALU13'から伝送されるデータに置き換える。そして、これらに、演算に供しない部分（すなわち、第2のメモリブロック21に格納されていた上位8ビットのデータ）を付加して出力データを生成し、それらをメモリ17に出力する。

【0053】

メモリ17は、混合回路29から伝送された出力データを元のアドレスに格納する。すなわち、出力データの下位16ビットを第1のメモリブロック19のアドレスnに格納し、上位16ビットを第2のメモリブロック21のアドレスn+1に格納する。

【0054】

その結果、パターン1により、第1及び第2のメモリブロック19、21に格

納されていたデータは、第2のメモリブロック21に格納されていた上位8ビットだけが元の値のままで、それ以外が新たな値に更新される。

【0055】

図13(b)に示すパターン2において、まず図示しない制御部は、第2のメモリブロック21からアドレス $n+1$ に格納されたデータを読み出すとともに、第1のメモリブロック19からアドレス $n+2$ に格納されたデータを読み出し、シフター11とレジスタ27に出力する。なお、アドレス $n+1$ に格納されたデータとは $R0'$ パートデータの一部と未使用データと $I1$ パートデータの一部である。またアドレス $n+2$ に格納されたデータとは $I1$ パートデータの一部と $R1$ パートデータの一部と未使用データである。

【0056】

シフター11は、アキュムレータ15やメモリ17から伝送されるデータを所定ビットシフトして位相の調整を行う。そして、その結果を $ALU13'$ に出力する。

【0057】

$ALU13'$ は、シフター11から伝送されるデータに対し、分割信号 K に基づいて、演算に供するデータと演算に供しないデータとに分割し、演算に供するデータを取得する。

【0058】

次に、 $ALU13'$ は、アキュムレータ15から前回サイクルの演算結果を取得する。そして、 $ALU13'$ は、シフター11から取得した演算に供するデータとアキュムレータ15から取得した前回サイクルの演算結果を用いて、DMFアルゴリズムによる演算を実行する。図13(b)では、このときの演算結果を、 $I1'$ パートデータと $R1'$ パートデータの組合せによるデータとして示している。

【0059】

$ALU13'$ は、演算結果をアキュムレータ15に出力する。アキュムレータ15は $ALU13'$ による演算結果を格納するとともに、演算結果を第1のルート23に沿って混合回路29に出力する。また、図示しない制御部の制御に基づ

いて所定のタイミングで演算結果を第2のルート25に沿ってALU13'に出力する。

【0060】

他方、レジスタ27は、図示しない制御部の制御に基づいて、所定のタイミングでメモリ17から伝送されたデータを第3のルート23'に沿って混合回路29に出力する。

【0061】

混合回路29は、図示しない制御部の制御に基づいて、第3のルート23'に沿ってレジスタ27から伝送されるデータを8ビット毎に分割し、その中の演算に供する部分（すなわち、第2のメモリブロック21に格納されていた上位8ビットと、第1のメモリブロック19に格納されていた上位8ビットと下位8ビットのデータ）を、第1のルート23に沿ってALU13'から伝送されるデータに置き換える。そして、これらに、演算に供しない部分（すなわち、第2のメモリブロック21に格納されていた下位8ビットのデータ）を付加して出力データを生成し、それらをメモリ17に出力する。

【0062】

メモリ17は、混合回路29から伝送される出力データを元のアドレスに格納する。すなわち、出力データの下位16ビットを第2のメモリブロック21のアドレス $n+1$ に格納し、上位16ビットを第1のメモリブロック19のアドレス $n+2$ に格納する。

【0063】

その結果、パターン2により、第1及び第2のメモリブロック19、21に格納されていたデータは、第2のメモリブロック21に格納されていた下位8ビットだけが元の値のままで、それ以外が新たな値に更新される。

【0064】

図13(c)に示すパターン3において、まず図示しない制御部は、第2のメモリブロック21からアドレス $n+3$ に格納されたデータを読み出すとともに、第1のメモリブロック19からアドレス $n+4$ に格納されたデータを読み出し、シフター11とレジスタ27に出力する。なお、アドレス $n+3$ に格納されたデ

ータとは I 2 パートデータと R 2 パートデータの一部である。またアドレス $n + 4$ に格納されたデータとは R 2 パートデータの一部と未使用データと I 3 パートデータである。

【 0 0 6 5 】

シフター 1 1 は、アキュムレータ 1 5 やメモリ 1 7 から伝送されるデータを所定ビットシフトして位相の調整を行う。そして、その結果を A L U 1 3 ' に出力する。

【 0 0 6 6 】

A L U 1 3 ' は、シフター 1 1 から伝送されるデータに対し、分割信号 K に基づいて、演算に供するデータと演算に供しないデータとに分割し、演算に供するデータを取得する。

【 0 0 6 7 】

次に、A L U 1 3 ' は、アキュムレータ 1 5 から前回サイクルの演算結果を取得する。そして、A L U 1 3 ' は、シフター 1 1 から取得した演算に供するデータとアキュムレータ 1 5 から取得した前回サイクルの演算結果を用いて、DMF アルゴリズムによる演算を実行する。図 1 3 (c) では、このときの演算結果を、I 2 ' パートデータと R 2 ' パートデータの組合せによるデータとして示している。

【 0 0 6 8 】

A L U 1 3 ' は、演算結果をアキュムレータ 1 5 に出力する。アキュムレータ 1 5 は A L U 1 3 ' による演算結果を格納するとともに、演算結果を第 1 のルート 2 3 に沿って混合回路 2 9 に出力する。また、図示しない制御部の制御に基づいて所定のタイミングで演算結果を第 2 のルート 2 5 に沿って A L U 1 3 ' に出力する。

【 0 0 6 9 】

他方、レジスタ 2 7 は、図示しない制御部の制御に基づいて、所定のタイミングでメモリ 1 7 から伝送されたデータを第 3 のルート 2 3 ' に沿って混合回路 2 9 に出力する。

【 0 0 7 0 】

混合回路 2 9 は、図示しない制御部の制御に基づいて、第 3 のルート 2 3' に沿ってレジスタ 2 7 から伝送されるデータを 8 ビット毎に分割し、その中の演算に供する部分（すなわち、第 2 のメモリブロック 2 1 に格納されていた下位 8 ビットと上位 8 ビットと、第 1 のメモリブロック 1 9 に格納されていた下位 8 ビットのデータ）を、第 1 のルート 2 3 に沿って A L U 1 3' から伝送されるデータに置き換える。そして、これらに、演算に供しない部分（すなわち、第 1 のメモリブロック 1 9 に格納されていた上位 8 ビットのデータ）を付加して出力データを生成し、それらをメモリ 1 7 に出力する。

【 0 0 7 1 】

メモリ 1 7 は、混合回路 2 9 から伝送される出力データを元のアドレスに格納する。すなわち、出力データの下位 1 6 ビットを第 2 のメモリブロック 2 1 のアドレス $n + 3$ に格納し、上位 1 6 ビットを第 1 のメモリブロック 1 9 のアドレス $n + 4$ に格納する。

【 0 0 7 2 】

その結果、パターン 3 により、第 1 及び第 2 のメモリブロック 1 9、2 1 に格納されていたデータは、第 1 のメモリブロック 1 9 に格納されていた上位 8 ビットだけが元の値のままで、それ以外が新たな値に更新される。

【 0 0 7 3 】

図 1 5 は混合回路 2 9 における出力データの遷移を示す図である。図 1 5 中、左側の上段の A c c [2 3 : 0] , R e g [7 : 0] と左側の下段の R e g [1 5 : 8] , A c c [2 3 : 0] は、混合回路 2 9 が A c c 1 5 とレジスタ 2 7 からの出力を混合することによって生成したデータである。また、右側の O u t [3 1 : 0] は、混合回路 2 9 がメモリ 1 7 から読み出されたデータの元のアドレスに応じて左側の上段の A c c [2 3 : 0] , R e g [7 : 0] または左側の下段の R e g [1 5 : 8] , A c c [2 3 : 0] のいずれかを選択してメモリ 1 7 に出力する出力データである。

【 0 0 7 4 】

ここで、A c c [x : y] はアキュムレータ 1 5 からの x ビット目から y ビット目の出力データを表現しており、R e g [x : y] はレジスタ 2 7 からの x ビ

ット目から y ビット目の出力データを表現している。例えば $A c c [23:0]$, $R e g [7:0]$ は、アキュムレータ15からの0ビット目～23ビット目までの24ビット幅の出力データとレジスタ27からの0ビット目～7ビット目までの8ビット幅の出力データの組合せを表現している。また $O u t [x:y]$ は混合回路29からの x ビット目から y ビット目の出力データを表現している。例えば $O u t [31:0]$ は混合回路29からの0ビット目～31ビット目までの32ビット幅の出力データを表現している。

【0075】

なお、図示しない制御部は、一演算処理毎に、アドレス383のI255パートデータとR255パートデータに、次の遅延値 D_1 の入力 A_0 を配置し、次の演算処理時において、アドレス383を新たなアドレス0と見なして処理を実行する。これは、メモリ17のアドレス0とアドレス383がリング状に繋がっていると想定した場合に、基準点を1つ左に進めて処理することと同様になる。そのため、DSPは、簡易な制御で、演算に用いるデータを好適に出力することができるようになる。

【0076】

このような制御について、図16を用いて以下に詳述する。図16はサイクリックなデータの配置を示す図である。図16中、斜線を付した領域はデータの入力箇所を示している。

【0077】

図示しない制御部は、メモリ17のDMFアルゴリズムによって定まる所定のアドレスに格納されたデータを、遅延値 D_1 、 D_2 、 D_3 、 D_4 、 D_5 、 D_6 、 D_7 、 D_8 の出力データとしてサイクリックに用いる。そして、一演算処理毎に、これらのデータは、各遅延値 $D_1 \sim D_8$ の出力データとして、上述のパターン1～3によって演算された演算結果に更新される。なお、最終の遅延値 D_8 の出力データは、次の演算処理時において、先頭の遅延値 D_0 の入力データとなる。このようにして、図22及び図23に示すDMFアルゴリズム1サイクルの演算処理が実行される。すると、図示しない制御部は、図16に示す基準点の位置を所定量（ここでは1個）だけ左回りに移動させて、次の演算処理を実行する。このときの

演算処理は、上述した動作と同様の動作となる。このようにして、図示しない制御部は、メモリ 17 から各遅延値に対応する入出力を次々と容易に読み出すことができる。このような機能は、DSP に標準的に設けられているモデュロアドレッシングを利用することにより簡単に実現できる。

【 0 0 7 8 】

以下に、DSP 内部における各構成の動作について説明する。なお、ALU 13' は、分割信号 K やレジスタ値に基づいて、キャリー信号 C を任意のビット位置で切断可能としている。本実施の形態では、ALU 13' は、10 ビット幅で切断している。

【 0 0 7 9 】

メモリ 17 に格納されたデータの構成が図 13 (a) に示すパターン 1 の場合、DSP 内部における各構成は以下のように動作する。

【 0 0 8 0 】

まず、図示しない制御部は、第 1 のメモリブロック 19 と第 2 のメモリブロック 21 の所定のアドレスから合計 32 ビット幅のデータを読み出し、シフター 11 とレジスタ 27 に出力する。シフター 11 は、データをシフトせずに、そのまま ALU 13' に出力する。また、レジスタ 27 は、第 1 のメモリブロック 19 と第 2 のメモリブロック 21 の所定のアドレスから読み出されたデータを一時格納する。

【 0 0 8 1 】

この後、ALU 13' は、演算処理を実行して、その演算結果を Acc_0 と Acc_1 に出力する。

【 0 0 8 2 】

Acc_0 と Acc_1 は、演算結果を一時格納した後、それをシフター 11 に出力する。シフター 11 は、ALU 13' を介して演算結果をそのまま Acc_0 と Acc_1 に出力する。 Acc_0 と Acc_1 は、再び演算結果を一時格納する。

【 0 0 8 3 】

この後、レジスタ 27 は格納していたデータを混合回路 29 に出力するとともに、 Acc_0 と Acc_1 は ALU 13' による演算結果を混合回路 29 に出力する

【 0 0 8 4 】

次に、混合回路 2 9 は、最下位ビット (LSB) 側の 8 ビットがレジスタ 2 7 に格納されていたデータとなるように、レジスタ 2 7 に格納されていたデータと Acc_0 と Acc_1 に格納されていた $ALU13'$ による演算結果とを混合し、その混合結果をメモリ 1 7 に出力する。メモリ 1 7 は、混合結果を第 1 のメモリブロック 1 9 と第 2 のメモリブロック 2 1 の元のアドレスに格納する。

【 0 0 8 5 】

メモリ 1 7 に格納されたデータの構成が図 1 3 (b) に示すパターン 2 の場合、DSP 内部における各構成は以下のように動作する。

【 0 0 8 6 】

まず、図示しない制御部は、第 2 のメモリブロック 2 1 と第 1 のメモリブロック 1 9 の所定のアドレスから合計 3 2 ビット幅のデータを読み出し、シフター 1 1 とレジスタ 2 7 に出力する。シフター 1 1 は、データを右に 8 ビットシフトして $ALU13'$ に出力する。また、レジスタ 2 7 は、第 2 のメモリブロック 2 1 と第 1 のメモリブロック 1 9 の所定のアドレスから読み出されたデータを一時格納する。

【 0 0 8 7 】

この後、 $ALU13'$ は、演算処理を実行して、その演算結果を Acc_0 と Acc_1 に出力する。 Acc_0 と Acc_1 は、演算結果を一時格納した後、それをシフター 1 1 に出力する。シフター 1 1 は、演算結果を左に 8 ビットシフトして $ALU13'$ を介して Acc_0 と Acc_1 に出力する。 Acc_0 と Acc_1 は、左に 8 ビットシフトされた $ALU13'$ による演算結果を一時格納する。

【 0 0 8 8 】

この後、レジスタ 2 7 は格納していたデータを混合回路 2 9 に出力するとともに、 Acc_0 と Acc_1 は左に 8 ビットシフトされた $ALU13'$ による演算結果を混合回路 2 9 に出力する。

【 0 0 8 9 】

次に、混合回路 2 9 は、最上位ビット (MSB) 側の 8 ビットがレジスタ 2 7

に格納されていたデータとなるように、レジスタ 27 に格納されていたデータと $A c c_0$ と $A c c_1$ に格納されていた $A L U 13'$ による演算結果とを混合し、その混合結果をメモリ 17 に出力する。メモリ 17 は、混合結果を第 2 のメモリブロック 21 と第 1 のメモリブロック 19 の元のアドレスに格納する。

【 0 0 9 0 】

メモリ 17 に格納されたデータの構成が図 13 (c) に示すパターン 3 の場合、DSP 内部における各構成は以下のように動作する。

【 0 0 9 1 】

まず、図示しない制御部は、第 2 のメモリブロック 21 と第 1 のメモリブロック 19 の所定のアドレスから合計 32 ビット幅のデータを読み出し、シフター 11 とレジスタ 27 に出力する。シフター 11 は、データをシフトせずに、そのまま $A L U 13'$ に出力する。また、レジスタ 27 は、第 2 のメモリブロック 21 と第 1 のメモリブロック 19 の所定のアドレスから読み出されたデータを一時格納する。

【 0 0 9 2 】

この後、 $A L U 13'$ は、演算処理を実行して、その演算結果を $A c c_0$ と $A c c_1$ に出力する。 $A c c_0$ と $A c c_1$ は、演算結果を一時格納した後、それをシフター 11 に出力する。シフター 11 は、 $A L U 13'$ を介して演算結果をそのまま $A c c_0$ と $A c c_1$ に出力する。 $A c c_0$ と $A c c_1$ は、再び演算結果を一時格納する。

【 0 0 9 3 】

この後、レジスタ 27 は格納していたデータを混合回路 29 に出力するとともに、 $A c c_0$ と $A c c_1$ は $A L U 13'$ による演算結果を混合回路 29 に出力する。

【 0 0 9 4 】

次に、混合回路 29 は、最下位ビット (LSB) 側の 8 ビットがレジスタ 27 に格納されていたデータとなるように、レジスタ 27 に格納されていたデータと $A c c_0$ と $A c c_1$ に格納されていた $A L U 13'$ による演算結果とを混合し、その混合結果をメモリ 17 に出力する。メモリ 17 は、混合結果を第 2 のメモリブ

ロック 21 と第 1 のメモリブロック 19 の元のアドレスに格納する。

【0095】

以下に、ALU13' 内部における演算過程を説明する。図 17 と図 18 は DMF アルゴリズムの概要を示す図である。図 17 と図 18 において括弧でくくる部分の演算過程を一例として詳述する。

【0096】

まず、図示しない制御部は、第 1 のメモリブロック 19 と第 2 のメモリブロック 21 のアドレス 0 と 1 から合計 32 ビット幅のデータを読み出し、シフター 11 とレジスタ 27 に出力する。シフター 11 は、データをシフトせずに、そのまま ALU13' に出力する。また、レジスタ 27 は、第 1 のメモリブロック 19 と第 2 のメモリブロック 21 のアドレス 0 と 1 から読み出されたデータを一時格納する。この後、ALU13' は、演算処理を実行する。

【0097】

その際に、まず、図示しない制御部がデータの構成に基づいて分割信号 K を生成し、ALU13' は分割信号 K に基づいて第 1 のメモリブロック 19 と第 2 のメモリブロック 21 のアドレス 0 と 1 から読み出されたデータを分割して入力 A_0 を取得する。次に、ALU13' は、図 17 に示すように、 A_1 、 B_1 、 C_1 、 C_1' を算出する。これらの値は、図 24 ～ 図 27 に示す通りである。このようにして算出された演算結果は、図 17 に示す DMF アルゴリズムの概要を示す図の上の段によるもの（例えば C_1 ）がアキュムレータ 15 の Acc_1 に格納され、下の段によるもの（例えば B_1 、 C_1' ）がアキュムレータ 15 の Acc_0 に格納される。このようにして演算結果は、アキュムレータ 15 の Acc_0 及び Acc_1 に格納される。

【0098】

特に、演算結果 C_1 は、アキュムレータ 15 の Acc_0 及び Acc_1 に格納された遅延値 D_1 に対応するデータの最後に上書きで書き込まれる。そして、演算結果 C_1 は、データ D_1 とデータ D_2 が連続しているため、次の遅延値 D_2 の入力データとなる。

【0099】

なお、第1のメモリブロック19と第2のメモリブロック21のアドレス0と1から読み出されたデータのうち、演算に供しない部分は、レジスタ27に格納されている。この部分は、混合回路29によって演算結果と混合される。混合回路29は、その混合結果をメモリ17に出力し、第1のメモリブロック19と第2のメモリブロック21のアドレス0と1に格納させる。そのため、本実施の形態は、演算に供しない部分を演算するための時間を必要とせず、簡易に出力データを生成することができる。また、上書き時に未使用データをそのまま書き出すこともできる。

【0100】

次に、ALU13'は、図18に示すように、 A_2 、 B_2 、 C_2 、 C_2' を算出する。これらの値は、図24～図27に示す通りである。このようにして算出された演算結果は、図18に示すDMFアルゴリズムの概要を示す図の上の段によるもの（例えば C_2 ）がアキュムレータ15の Acc_1 に格納され、下の段によるもの（例えば B_2 、 C_2' ）がアキュムレータ15の Acc_0 に格納される。このようにして演算結果は、アキュムレータ15の Acc_0 及び Acc_1 に格納される。

【0101】

特に、演算結果 C_2 は、アキュムレータ15の Acc_0 及び Acc_1 に格納された遅延値 D_2 に対応するデータの最後に上書きで書き込まれる。そして、演算結果 C_2 は、データ D_2 とデータ D_3 が連続しているため、次の遅延値 D_3 の入力データとなる。

【0102】

このようにして、ALU13'は、同様の演算を連続して実行し、最終的な演算結果を得る。

【0103】

その後、図示しない制御部は、図16に示す基準点の位置を、1個だけ左回りに移動させ、メモリ17に格納するデータをサイクリックに1個前進した形にして、次の演算処理を実行する。

【0104】

以上詳細に説明したように、本発明は、所定ビット幅のレジスタ27と混合回

路 2 9 を設けるだけで、DMF 処理における使用メモリ量を低減できるという効果を有する。そのため、例えば第 3 世代携帯電話に用いられる DSP に適用した場合に、メモリ量を、510 ワードから 384 ワードに、すなわち、126 ワード分低減をさせることができる。

【0105】

しかも、本発明は、大幅な回路の増大がないので、極めて少ないコストでこのような効果を達成できる。

【0106】

本発明は上述の実施の形態に限定されることなく、本発明の要旨を逸脱しない範囲で種々の応用及び変形が考えられる。例えば、本実施の形態においては、メモリ 17 を 32 ビット幅のメモリによって構成してもよい。また、本発明は、DSP だけでなく、9 ～ 12 ビット幅を有するデータの処理を行う機器に対しても適用することができる。

【0107】

【発明の効果】

以上説明した本発明は、所定ビット幅のレジスタ 27 と混合回路 29 を設けるだけで、DMF 処理における使用メモリ量を低減できるという効果を有する。

【図面の簡単な説明】

【図 1】

本発明に係る演算装置の構成を示す図である。

【図 2】

2 つのメモリブロックに格納されるデータの配置を示す図である。

【図 3】

2 つのメモリブロックに格納されるデータの配置を示す図である。

【図 4】

2 つのメモリブロックに格納されるデータの配置を示す図である。

【図 5】

2 つのメモリブロックに格納されるデータの配置を示す図である。

【図 6】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 7】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 8】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 9】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 1 0】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 1 1】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 1 2】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 1 3】

2つのメモリブロックに格納されるデータの遷移過程を示す図である。

【図 1 4】

本実施の形態に係る A L U の内部構成を示す図である。

【図 1 5】

混合回路における出力の遷移を示す図である。

【図 1 6】

サイクリックなデータの配置を示す図である。

【図 1 7】

D M F アルゴリズムの概要を示す図である。

【図 1 8】

D M F アルゴリズムの概要を示す図である。

【図 1 9】

従来の演算装置の構成を示す図である。

【図 2 0】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 2 1】

2つのメモリブロックに格納されるデータの配置を示す図である。

【図 2 2】

DMF アルゴリズムの概要を示す図である。

【図 2 3】

DMF アルゴリズムの概要を示す図である。

【図 2 4】

DMF アルゴリズムの算術式を示す図表である。

【図 2 5】

DMF アルゴリズムの算術式を示す図表である。

【図 2 6】

DMF アルゴリズムの算術式を示す図表である。

【図 2 7】

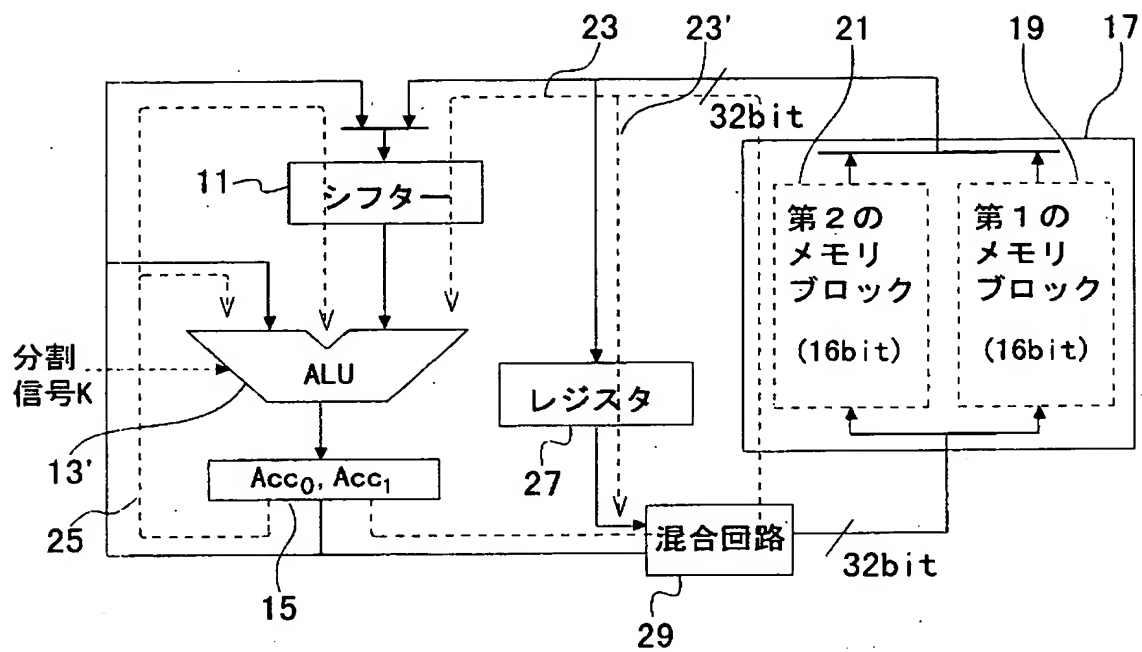
DMF アルゴリズムの算術式を示す図表である。

【符号の説明】

- 1 1 シフター
- 1 3' ALU
- 1 5 アキュムレータ
- 1 7 メモリ
- 1 9 第 1 のメモリブロック
- 2 1 第 2 のメモリブロック
- 2 3 第 1 のルート
- 2 3' 第 3 のルート
- 2 5 第 2 のルート
- 2 7 レジスタ
- 2 9 混合回路

【書類名】 図面

【図 1】



本発明に係る演算装置の構成を示す図

【図 2】

アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n	R0				I0											
n + 2	未使用データ				R1				I1							
n + 4					I3				未使用データ				R2			
n + 6					R4				I4							

第 1 のメモリブロックのデータ

アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n + 1	I1				R0											
n + 3					R2				I2							
n + 5	未使用データ								R3				I3			
n + 7					I5				未使用データ				R4			

第 2 のメモリブロックのデータ

2 つのメモリブロックに格納されるデータの配置を示す図

【図 3】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1	-	-	-	1	-	-	-	-	-	*	*	*	*	*	-	-
3	R	2	-	-	-	-	-	-	-	-	-	1	2	-	-	-
5	*	*	*	*	*	*	-	-	R	3	-	-	-	-	-	-
7	-	-	-	1	5	-	-	-	-	*	*	*	*	*	-	-
9	R	6	-	-	-	-	-	-	-	-	-	1	6	-	-	-
11	*	*	*	*	*	*	-	-	R	7	-	-	-	-	-	-
13	-	-	-	1	9	-	-	-	-	*	*	*	*	*	-	-
15	R	1	0	-	-	-	-	-	-	-	-	1	1	0	-	-
17	*	*	*	*	*	*	-	-	R	1	1	-	-	-	-	-
19	-	-	-	1	1	3	-	-	-	*	*	*	*	*	-	-
21	R	1	4	-	-	-	-	-	-	-	-	1	1	4	-	-
23	*	*	*	*	*	*	-	-	R	1	5	-	-	-	-	-
25	-	-	-	1	1	7	-	-	-	*	*	*	*	*	-	-
27	R	1	8	-	-	-	-	-	-	-	-	1	1	8	-	-
29	*	*	*	*	*	*	-	-	R	1	9	-	-	-	-	-
31	-	-	-	1	2	1	-	-	-	*	*	*	*	*	-	-
33	R	2	2	-	-	-	-	-	-	-	-	1	2	2	-	-
35	*	*	*	*	*	*	-	-	R	2	3	-	-	-	-	-
37	-	-	-	1	2	5	-	-	-	*	*	*	*	*	-	-
39	R	2	6	-	-	-	-	-	-	-	-	-	1	2	6	-

第2のメモリブロックのデータ

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	R	0	-	-	-	-	-	-	-	-	-	1	0	-	-	-
2	*	*	*	*	*	*	-	-	R	1	-	-	-	-	-	-
4	-	-	-	1	3	-	-	-	-	*	*	*	*	*	-	-
6	R	4	-	-	-	-	-	-	-	-	-	1	4	-	-	-
8	*	*	*	*	*	*	-	-	R	5	-	-	-	-	-	-
10	-	-	-	1	7	-	-	-	-	*	*	*	*	*	-	-
12	R	8	-	-	-	-	-	-	-	-	-	1	8	-	-	-
14	*	*	*	*	*	*	-	-	R	9	-	-	-	-	-	-
16	-	-	-	1	1	1	-	-	-	*	*	*	*	*	-	-
18	R	1	2	-	-	-	-	-	-	-	-	1	1	2	-	-
20	*	*	*	*	*	*	-	-	R	1	3	-	-	-	-	-
22	-	-	-	1	1	5	-	-	-	*	*	*	*	*	-	-
24	R	1	6	-	-	-	-	-	-	-	-	1	1	6	-	-
26	*	*	*	*	*	*	-	-	R	1	7	-	-	-	-	-
28	-	-	-	1	1	9	-	-	-	*	*	*	*	*	-	-
30	R	2	0	-	-	-	-	-	-	-	-	1	2	0	-	-
32	*	*	*	*	*	*	-	-	R	2	1	-	-	-	-	-
34	-	-	-	1	2	3	-	-	-	*	*	*	*	*	-	-
36	R	2	4	-	-	-	-	-	-	-	-	1	2	4	-	-
38	*	*	*	*	*	*	-	-	R	2	5	-	-	-	-	-

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図4】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
40	-	-	-	1	2	7	-	-	-	*	*	*	*	-	-	-
42	R	2	8	-	-	-	-	-	-	-	-	-	1	2	8	-
44	*	*	*	*	*	-	-	-	-	R	2	9	-	-	-	-
46	-	-	-	1	3	1	-	-	-	*	*	*	*	-	-	-
48	R	3	2	-	-	-	-	-	-	-	-	-	1	3	2	-
50	*	*	*	*	*	-	-	-	-	R	3	3	-	-	-	-
52	-	-	-	1	3	5	-	-	-	*	*	*	*	-	-	-
54	R	3	6	-	-	-	-	-	-	-	-	-	1	3	6	-
56	*	*	*	*	*	-	-	-	-	R	3	7	-	-	-	-
58	-	-	-	1	3	9	-	-	-	*	*	*	*	-	-	-
60	R	4	0	-	-	-	-	-	-	-	-	-	1	4	0	-
62	*	*	*	*	*	-	-	-	-	R	4	1	-	-	-	-
64	-	-	-	1	4	3	-	-	-	*	*	*	*	-	-	-
66	R	4	4	-	-	-	-	-	-	-	-	-	1	4	4	-
68	*	*	*	*	*	-	-	-	-	R	4	5	-	-	-	-
70	-	-	-	1	4	7	-	-	-	*	*	*	*	-	-	-
72	R	4	8	-	-	-	-	-	-	-	-	-	1	4	8	-
74	*	*	*	*	*	-	-	-	-	R	4	9	-	-	-	-
76	-	-	-	1	5	1	-	-	-	*	*	*	*	-	-	-
78	R	5	2	-	-	-	-	-	-	-	-	-	1	5	2	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
41	*	*	*	*	-	-	-	-	-	R	2	7	-	-	-	-
43	-	-	-	1	2	9	-	-	-	*	*	*	*	-	-	-
45	R	3	0	-	-	-	-	-	-	-	-	1	3	0	-	-
47	*	*	*	*	-	-	-	-	-	R	3	1	-	-	-	-
49	-	-	-	1	3	3	-	-	-	*	*	*	*	-	-	-
51	R	3	4	-	-	-	-	-	-	-	-	1	3	4	-	-
53	*	*	*	*	-	-	-	-	-	R	3	5	-	-	-	-
55	-	-	-	1	3	7	-	-	-	*	*	*	*	-	-	-
57	R	3	8	-	-	-	-	-	-	-	-	1	3	8	-	-
59	*	*	*	*	-	-	-	-	-	R	3	9	-	-	-	-
61	-	-	-	1	4	1	-	-	-	*	*	*	*	-	-	-
63	R	4	2	-	-	-	-	-	-	-	-	1	4	2	-	-
65	*	*	*	*	-	-	-	-	-	R	4	3	-	-	-	-
67	-	-	-	1	4	5	-	-	-	*	*	*	*	-	-	-
69	R	4	6	-	-	-	-	-	-	-	-	1	4	6	-	-
71	*	*	*	*	-	-	-	-	-	R	4	7	-	-	-	-
73	-	-	-	1	4	9	-	-	-	*	*	*	*	-	-	-
75	R	5	0	-	-	-	-	-	-	-	-	1	5	0	-	-
77	*	*	*	*	-	-	-	-	-	R	5	1	-	-	-	-
79	-	-	-	1	5	3	-	-	-	*	*	*	*	-	-	-

第2のメモリブロックのデータ

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 5】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
80	*	*	*	*	*	*	-	-	-	-	R	5	3	-	-	-
82	-	-	-	1	5	5	-	-	-	-	*	*	*	*	-	-
84	R	5	6	-	-	-	-	-	-	-	-	-	1	5	6	-
86	*	*	*	*	*	*	-	-	-	-	R	5	7	-	-	-
88	-	-	1	5	9	-	-	-	-	-	*	*	*	*	-	-
90	R	6	0	-	-	-	-	-	-	-	-	-	1	6	0	-
92	*	*	*	*	*	*	-	-	-	-	R	6	1	-	-	-
94	-	-	1	6	3	-	-	-	-	-	*	*	*	*	-	-
96	R	6	4	-	-	-	-	-	-	-	-	-	1	6	4	-
98	*	*	*	*	*	*	-	-	-	-	R	6	5	-	-	-
100	-	-	1	6	7	-	-	-	-	-	*	*	*	*	-	-
102	R	6	8	-	-	-	-	-	-	-	-	-	1	6	8	-
104	*	*	*	*	*	*	-	-	-	-	R	6	9	-	-	-
106	-	-	1	7	1	-	-	-	-	-	*	*	*	*	-	-
108	R	7	2	-	-	-	-	-	-	-	-	-	1	7	2	-
110	*	*	*	*	*	*	-	-	-	-	R	7	3	-	-	-
112	-	-	1	7	5	-	-	-	-	-	*	*	*	*	-	-
114	R	7	6	-	-	-	-	-	-	-	-	-	1	7	6	-
116	*	*	*	*	*	*	-	-	-	-	R	7	7	-	-	-
118	-	-	1	7	9	-	-	-	-	-	*	*	*	*	-	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
81	R	5	4	-	-	-	-	-	-	-	1	5	4	-	-	-
83	*	*	*	*	*	*	-	-	-	R	5	5	-	-	-	-
85	-	-	1	5	7	-	-	-	-	*	*	*	*	-	-	-
87	R	5	8	-	-	-	-	-	-	-	1	5	8	-	-	-
89	*	*	*	*	*	*	-	-	-	R	5	9	-	-	-	-
91	-	-	1	6	1	-	-	-	-	*	*	*	*	-	-	-
93	R	6	2	-	-	-	-	-	-	-	1	6	2	-	-	-
95	*	*	*	*	*	*	-	-	-	R	6	3	-	-	-	-
97	-	-	1	6	5	-	-	-	-	*	*	*	*	-	-	-
99	R	6	6	-	-	-	-	-	-	-	1	6	6	-	-	-
101	*	*	*	*	*	*	-	-	-	R	6	7	-	-	-	-
103	-	-	1	6	9	-	-	-	-	*	*	*	*	-	-	-
105	R	7	0	-	-	-	-	-	-	-	1	7	0	-	-	-
107	*	*	*	*	*	*	-	-	-	R	7	1	-	-	-	-
109	-	-	1	7	3	-	-	-	-	*	*	*	*	-	-	-
111	R	7	4	-	-	-	-	-	-	-	1	7	4	-	-	-
113	*	*	*	*	*	*	-	-	-	R	7	5	-	-	-	-
115	-	-	1	7	7	-	-	-	-	*	*	*	*	-	-	-
117	R	7	8	-	-	-	-	-	-	-	1	7	8	-	-	-
119	*	*	*	*	*	*	-	-	-	R	7	9	-	-	-	-

第2のメモリブロックのデータ

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 6】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
121	-	-	-	1	8	1	-	-	-	*	*	*	*	-	-	-
123	R	8	2	-	-	-	-	-	-	-	-	1	8	2	-	-
125	*	*	*	*	*	-	-	-	R	8	3	-	-	-	-	-
127	-	-	-	1	8	5	-	-	-	*	*	*	*	-	-	-
129	R	8	6	-	-	-	-	-	-	-	-	1	8	6	-	-
131	*	*	*	*	*	-	-	-	R	8	7	-	-	-	-	-
133	-	-	-	1	8	9	-	-	-	*	*	*	*	-	-	-
135	R	9	0	-	-	-	-	-	-	-	-	1	9	0	-	-
137	*	*	*	*	*	-	-	-	R	9	1	-	-	-	-	-
139	-	-	-	1	9	3	-	-	-	*	*	*	*	-	-	-
141	R	9	4	-	-	-	-	-	-	-	-	1	9	4	-	-
143	*	*	*	*	*	-	-	-	R	9	5	-	-	-	-	-
145	-	-	-	1	9	7	-	-	-	*	*	*	*	-	-	-
147	R	9	8	-	-	-	-	-	-	-	-	1	9	8	-	-
149	*	*	*	*	*	-	-	-	R	9	9	-	-	-	-	-
151	-	-	-	1	1	0	1	-	-	*	*	*	*	-	-	-
153	R	1	0	2	-	-	-	-	-	-	-	1	0	2	-	-
155	*	*	*	*	*	-	-	-	R	1	0	3	-	-	-	-
157	-	-	-	1	1	0	5	-	-	*	*	*	*	-	-	-
159	R	1	0	6	-	-	-	-	-	-	-	1	0	6	-	-

第2のメモリブロックのデータ

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 7】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
160	-	-	-	-	1	0	7	-	*	*	*	*	*	*	-	-
162	R	1	0	8	-	-	-	-	-	-	-	-	-	1	0	8
164	*	*	*	*	*	*	-	-	-	R	1	0	9	-	-	-
166	-	-	-	-	1	1	1	-	*	*	*	*	*	*	-	-
168	R	1	1	2	-	-	-	-	-	-	-	-	1	1	2	-
170	*	*	*	*	*	*	-	-	-	R	1	1	3	-	-	-
172	-	-	-	-	1	1	5	-	*	*	*	*	*	*	-	-
174	R	1	1	6	-	-	-	-	-	-	-	-	1	1	6	-
176	*	*	*	*	*	*	-	-	-	R	1	1	7	-	-	-
178	-	-	-	-	1	1	9	-	*	*	*	*	*	*	-	-
180	R	1	2	0	-	-	-	-	-	-	-	-	1	2	0	-
182	*	*	*	*	*	*	-	-	-	R	1	2	1	-	-	-
184	-	-	-	-	1	2	3	-	*	*	*	*	*	*	-	-
186	R	1	2	4	-	-	-	-	-	-	-	-	1	2	4	-
188	*	*	*	*	*	*	-	-	-	R	1	2	5	-	-	-
190	-	-	-	-	1	2	7	-	*	*	*	*	*	*	-	-
192	R	1	2	8	-	-	-	-	-	-	-	-	1	2	8	-
194	*	*	*	*	*	*	-	-	-	R	1	2	9	-	-	-
196	-	-	-	-	1	3	1	-	*	*	*	*	*	*	-	-
198	R	1	3	2	-	-	-	-	-	-	-	-	1	3	2	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
161	*	*	*	*	*	-	-	-	R	1	0	7	-	-	-	-
163	-	-	-	-	1	0	9	-	*	*	*	*	*	*	-	-
165	R	1	1	0	-	-	-	-	-	-	-	-	1	1	0	-
167	*	*	*	*	*	*	-	-	-	R	1	1	1	-	-	-
169	-	-	-	-	1	1	3	-	*	*	*	*	*	*	-	-
171	R	1	1	4	-	-	-	-	-	-	-	-	1	1	4	-
173	*	*	*	*	*	*	-	-	-	R	1	1	5	-	-	-
175	-	-	-	-	1	1	7	-	*	*	*	*	*	*	-	-
177	R	1	1	8	-	-	-	-	-	-	-	-	1	1	8	-
179	*	*	*	*	*	*	-	-	-	R	1	1	9	-	-	-
181	-	-	-	-	1	2	1	-	*	*	*	*	*	*	-	-
183	R	1	2	2	-	-	-	-	-	-	-	-	1	2	2	-
185	*	*	*	*	*	*	-	-	-	R	1	2	3	-	-	-
187	-	-	-	-	1	2	5	-	*	*	*	*	*	*	-	-
189	R	1	2	6	-	-	-	-	-	-	-	-	1	2	6	-
191	*	*	*	*	*	*	-	-	-	R	1	2	7	-	-	-
193	-	-	-	-	1	2	9	-	*	*	*	*	*	*	-	-
195	R	1	3	0	-	-	-	-	-	-	-	-	1	3	0	-
197	*	*	*	*	*	*	-	-	-	R	1	3	1	-	-	-
199	-	-	-	-	1	3	3	-	*	*	*	*	*	*	-	-

第2のメモリブロックのデータ

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図8】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
200	*	*	*	*	*	*	-	-	-	-	R	1	3	3	-	-
202	-	-	1	1	3	5	-	-	-	-	*	*	*	*	-	-
204	R	1	3	6	-	-	-	-	-	-	R	1	3	6	-	-
206	*	*	*	*	*	*	-	-	-	-	R	1	3	7	-	-
208	-	-	1	1	3	9	-	-	-	-	*	*	*	*	-	-
210	R	1	4	0	-	-	-	-	-	-	-	-	1	4	0	-
212	*	*	*	*	*	*	-	-	-	-	R	1	4	1	-	-
214	-	-	1	1	4	3	-	-	-	-	*	*	*	*	-	-
216	R	1	4	4	-	-	-	-	-	-	-	-	1	4	4	-
218	*	*	*	*	*	*	-	-	-	-	R	1	4	5	-	-
220	-	-	1	1	4	7	-	-	-	-	*	*	*	*	-	-
222	R	1	4	8	-	-	-	-	-	-	-	-	1	4	8	-
224	*	*	*	*	*	*	-	-	-	-	R	1	4	9	-	-
226	-	-	1	1	5	1	-	-	-	-	*	*	*	*	-	-
228	R	1	5	2	-	-	-	-	-	-	-	-	1	5	2	-
230	*	*	*	*	*	*	-	-	-	-	R	1	5	3	-	-
232	-	-	1	1	5	5	-	-	-	-	*	*	*	*	-	-
234	R	1	5	6	-	-	-	-	-	-	-	-	1	5	6	-
236	*	*	*	*	*	*	-	-	-	-	R	1	5	7	-	-
238	-	-	1	1	5	9	-	-	-	-	*	*	*	*	-	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
201	R	1	3	4	-	-	-	-	-	-	-	1	3	4	-	-
203	*	*	*	*	*	*	-	-	-	-	R	1	3	5	-	-
205	-	-	1	1	3	7	-	-	-	-	*	*	*	*	-	-
207	R	1	3	8	-	-	-	-	-	-	-	1	3	8	-	-
209	*	*	*	*	*	*	-	-	-	-	R	1	3	9	-	-
211	-	-	1	1	4	1	-	-	-	-	*	*	*	*	-	-
213	R	1	4	2	-	-	-	-	-	-	-	1	4	2	-	-
215	*	*	*	*	*	*	-	-	-	-	R	1	4	3	-	-
217	-	-	1	1	4	5	-	-	-	-	*	*	*	*	-	-
219	R	1	4	6	-	-	-	-	-	-	-	1	4	6	-	-
221	*	*	*	*	*	*	-	-	-	-	R	1	4	7	-	-
223	-	-	1	1	4	9	-	-	-	-	*	*	*	*	-	-
225	R	1	5	0	-	-	-	-	-	-	-	1	5	0	-	-
227	*	*	*	*	*	*	-	-	-	-	R	1	5	1	-	-
229	-	-	1	1	5	3	-	-	-	-	*	*	*	*	-	-
231	R	1	5	4	-	-	-	-	-	-	-	1	5	4	-	-
233	*	*	*	*	*	*	-	-	-	-	R	1	5	5	-	-
235	-	-	1	1	5	7	-	-	-	-	*	*	*	*	-	-
237	R	1	5	8	-	-	-	-	-	-	-	1	5	8	-	-
239	*	*	*	*	*	*	-	-	-	-	R	1	5	9	-	-

第2のメモリブロックのデータ

第1のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 9】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
240	R	1	6	0	-	-	-	-	-	-	-	-	1	6	0	-
242	*	*	*	*	*	-	-	-	R	1	6	1	-	-	-	-
244	-	-	1	6	3	-	-	-	*	*	*	*	-	-	-	-
246	R	1	6	4	-	-	-	-	-	-	-	1	6	4	-	-
248	*	*	*	*	*	-	-	-	R	1	6	5	-	-	-	-
250	-	-	1	6	7	-	-	-	*	*	*	*	-	-	-	-
252	R	1	6	8	-	-	-	-	-	-	-	1	6	8	-	-
254	*	*	*	*	*	-	-	-	R	1	6	9	-	-	-	-
256	-	-	1	7	1	-	-	-	*	*	*	*	-	-	-	-
258	R	1	7	2	-	-	-	-	-	-	-	1	7	2	-	-
260	*	*	*	*	*	-	-	-	R	1	7	3	-	-	-	-
262	-	-	1	7	5	-	-	-	*	*	*	*	-	-	-	-
264	R	1	7	6	-	-	-	-	-	-	-	1	7	6	-	-
266	*	*	*	*	*	-	-	-	R	1	7	7	-	-	-	-
268	-	-	1	7	9	-	-	-	*	*	*	*	-	-	-	-
270	R	1	8	0	-	-	-	-	-	-	-	1	8	0	-	-
272	*	*	*	*	*	-	-	-	R	1	8	1	-	-	-	-
274	-	-	1	8	3	-	-	-	*	*	*	*	-	-	-	-
276	R	1	8	4	-	-	-	-	-	-	-	1	8	4	-	-
278	*	*	*	*	*	-	-	-	R	1	8	5	-	-	-	-

第1のメモリブロックのデータ

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
241	-	-	1	6	1	-	-	-	*	*	*	*	-	-	-	-
243	R	1	6	2	-	-	-	-	-	-	1	6	2	-	-	-
245	*	*	*	*	*	-	-	-	R	1	6	3	-	-	-	-
247	-	-	1	6	5	-	-	-	*	*	*	*	-	-	-	-
249	R	1	6	6	-	-	-	-	-	-	1	6	6	-	-	-
251	*	*	*	*	*	-	-	-	R	1	6	7	-	-	-	-
253	-	-	1	6	9	-	-	-	*	*	*	*	-	-	-	-
255	R	1	7	0	-	-	-	-	-	-	1	7	0	-	-	-
257	*	*	*	*	*	-	-	-	R	1	7	1	-	-	-	-
259	-	-	1	7	3	-	-	-	*	*	*	*	-	-	-	-
261	R	1	7	4	-	-	-	-	-	-	1	7	4	-	-	-
263	*	*	*	*	*	-	-	-	R	1	7	5	-	-	-	-
265	-	-	1	7	7	-	-	-	*	*	*	*	-	-	-	-
267	R	1	7	8	-	-	-	-	-	-	1	7	8	-	-	-
269	*	*	*	*	*	-	-	-	R	1	7	9	-	-	-	-
271	-	-	1	8	1	-	-	-	*	*	*	*	-	-	-	-
273	R	1	8	2	-	-	-	-	-	-	1	8	2	-	-	-
275	*	*	*	*	*	-	-	-	R	1	8	3	-	-	-	-
277	-	-	1	8	5	-	-	-	*	*	*	*	-	-	-	-
279	R	1	8	6	-	-	-	-	-	-	1	8	6	-	-	-

第2のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 10】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
280	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
282	R	1	8	8	-	-	-	-	-	-	-	-	-	-	-	-
284	*	*	*	*	*	-	-	-	-	-	R	1	8	9	-	-
286	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
288	R	1	9	2	-	-	-	-	-	-	-	-	-	-	-	-
290	*	*	*	*	*	-	-	-	-	-	R	1	9	3	-	-
292	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
294	R	1	9	6	-	-	-	-	-	-	-	-	-	-	-	-
296	*	*	*	*	*	-	-	-	-	-	R	1	9	7	-	-
298	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
300	R	2	0	0	-	-	-	-	-	-	-	-	-	-	-	-
302	*	*	*	*	*	-	-	-	-	-	R	2	0	1	-	-
304	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
306	R	2	0	4	-	-	-	-	-	-	-	-	-	-	-	-
308	*	*	*	*	*	-	-	-	-	-	R	2	0	5	-	-
310	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
312	R	2	0	8	-	-	-	-	-	-	-	-	-	-	-	-
314	*	*	*	*	*	-	-	-	-	-	R	2	0	9	-	-
316	-	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-
318	R	2	1	2	-	-	-	-	-	-	-	-	-	-	-	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
281	*	*	*	*	*	-	-	-	-	R	1	8	7	-	-	-
283	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
285	R	1	9	0	-	-	-	-	-	-	-	-	-	-	-	-
287	*	*	*	*	*	-	-	-	-	R	1	9	1	-	-	-
289	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
291	R	1	9	4	-	-	-	-	-	-	-	-	-	-	-	-
293	*	*	*	*	*	-	-	-	-	R	1	9	5	-	-	-
295	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
297	R	1	9	8	-	-	-	-	-	-	-	-	-	-	-	-
299	*	*	*	*	*	-	-	-	-	R	1	9	9	-	-	-
301	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
303	R	2	0	2	-	-	-	-	-	-	-	-	-	-	-	-
305	*	*	*	*	*	-	-	-	-	R	2	0	3	-	-	-
307	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
309	R	2	0	6	-	-	-	-	-	-	-	-	-	-	-	-
311	*	*	*	*	*	-	-	-	-	R	2	0	7	-	-	-
313	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-
315	R	2	1	0	-	-	-	-	-	-	-	-	-	-	-	-
317	*	*	*	*	*	-	-	-	-	R	2	1	1	-	-	-
319	-	-	-	-	-	-	-	-	-	*	*	*	*	-	-	-

第1のメモリーブロックのデータ

第2のメモリーブロックのデータ

2つのメモリーブロックに格納されるデータの配置を示す図

【図 11】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
320	*	*	*	*	*	*	-	-	-	R	2	1	3	-	-	-
322	-	-	1	2	1	5	-	-	-	*	*	*	*	-	-	-
324	R	2	1	6	-	-	-	-	-	-	-	1	2	1	6	-
326	*	*	*	*	*	*	-	-	-	R	2	1	7	-	-	-
328	-	-	1	2	1	9	-	-	-	*	*	*	*	-	-	-
330	R	2	2	0	-	-	-	-	-	-	-	1	2	2	0	-
332	*	*	*	*	*	*	-	-	-	R	2	2	1	-	-	-
334	-	-	1	2	2	3	-	-	-	*	*	*	*	-	-	-
336	R	2	2	4	-	-	-	-	-	-	-	1	2	2	4	-
338	*	*	*	*	*	*	-	-	-	R	2	2	5	-	-	-
340	-	-	1	2	2	7	-	-	-	*	*	*	*	-	-	-
342	R	2	2	8	-	-	-	-	-	-	-	1	2	2	8	-
344	*	*	*	*	*	*	-	-	-	R	2	2	9	-	-	-
346	-	-	1	2	3	1	-	-	-	*	*	*	*	-	-	-
348	R	2	3	2	-	-	-	-	-	-	-	1	2	3	2	-
350	*	*	*	*	*	*	-	-	-	R	2	3	3	-	-	-
352	-	-	1	2	3	5	-	-	-	*	*	*	*	-	-	-
354	R	2	3	6	-	-	-	-	-	-	-	1	2	3	6	-
356	*	*	*	*	*	*	-	-	-	R	2	3	7	-	-	-
358	-	-	1	2	3	9	-	-	-	*	*	*	*	-	-	-

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
321	R	2	1	4	-	-	-	-	-	-	1	2	1	4	-	-
323	*	*	*	*	*	*	-	-	-	R	2	1	5	-	-	-
325	-	-	1	2	1	7	-	-	-	*	*	*	*	-	-	-
327	R	2	1	8	-	-	-	-	-	-	1	2	1	8	-	-
329	*	*	*	*	*	*	-	-	-	R	2	1	9	-	-	-
331	-	-	1	2	2	1	-	-	-	*	*	*	*	-	-	-
333	R	2	2	2	-	-	-	-	-	-	1	2	2	2	-	-
335	*	*	*	*	*	*	-	-	-	R	2	2	3	-	-	-
337	-	-	1	2	2	5	-	-	-	*	*	*	*	-	-	-
339	R	2	2	6	-	-	-	-	-	-	1	2	2	6	-	-
341	*	*	*	*	*	*	-	-	-	R	2	2	7	-	-	-
343	-	-	1	2	2	9	-	-	-	*	*	*	*	-	-	-
345	R	2	3	0	-	-	-	-	-	-	1	2	3	0	-	-
347	*	*	*	*	*	*	-	-	-	R	2	3	1	-	-	-
349	-	-	1	2	3	3	-	-	-	*	*	*	*	-	-	-
351	R	2	3	4	-	-	-	-	-	-	1	2	3	4	-	-
353	*	*	*	*	*	*	-	-	-	R	2	3	5	-	-	-
355	-	-	1	2	3	7	-	-	-	*	*	*	*	-	-	-
357	R	2	3	8	-	-	-	-	-	-	1	2	3	8	-	-
359	*	*	*	*	*	*	-	-	-	R	2	3	9	-	-	-

第1のメモリブロックのデータ

第2のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 1 2】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
360	R	2	4	0	-	-	-	-	-	-	-	1	2	4	0	-
362	*	*	*	*	-	-	-	-	R	2	4	1	-	-	-	-
364	-	-	1	2	4	3	-	-	*	*	*	*	-	-	-	-
366	R	2	4	4	-	-	-	-	-	-	1	2	4	4	-	-
368	*	*	*	*	-	-	-	-	R	2	4	5	-	-	-	-
370	-	-	1	2	4	7	-	-	*	*	*	*	-	-	-	-
372	R	2	4	8	-	-	-	-	-	-	1	2	4	8	-	-
374	*	*	*	*	-	-	-	-	R	2	4	9	-	-	-	-
376	-	-	1	2	5	1	-	-	*	*	*	*	-	-	-	-
378	R	2	5	2	-	-	-	-	-	-	1	2	5	2	-	-
380	*	*	*	*	-	-	-	-	R	2	5	3	-	-	-	-
382	-	-	1	2	5	5	-	-	*	*	*	*	-	-	-	-

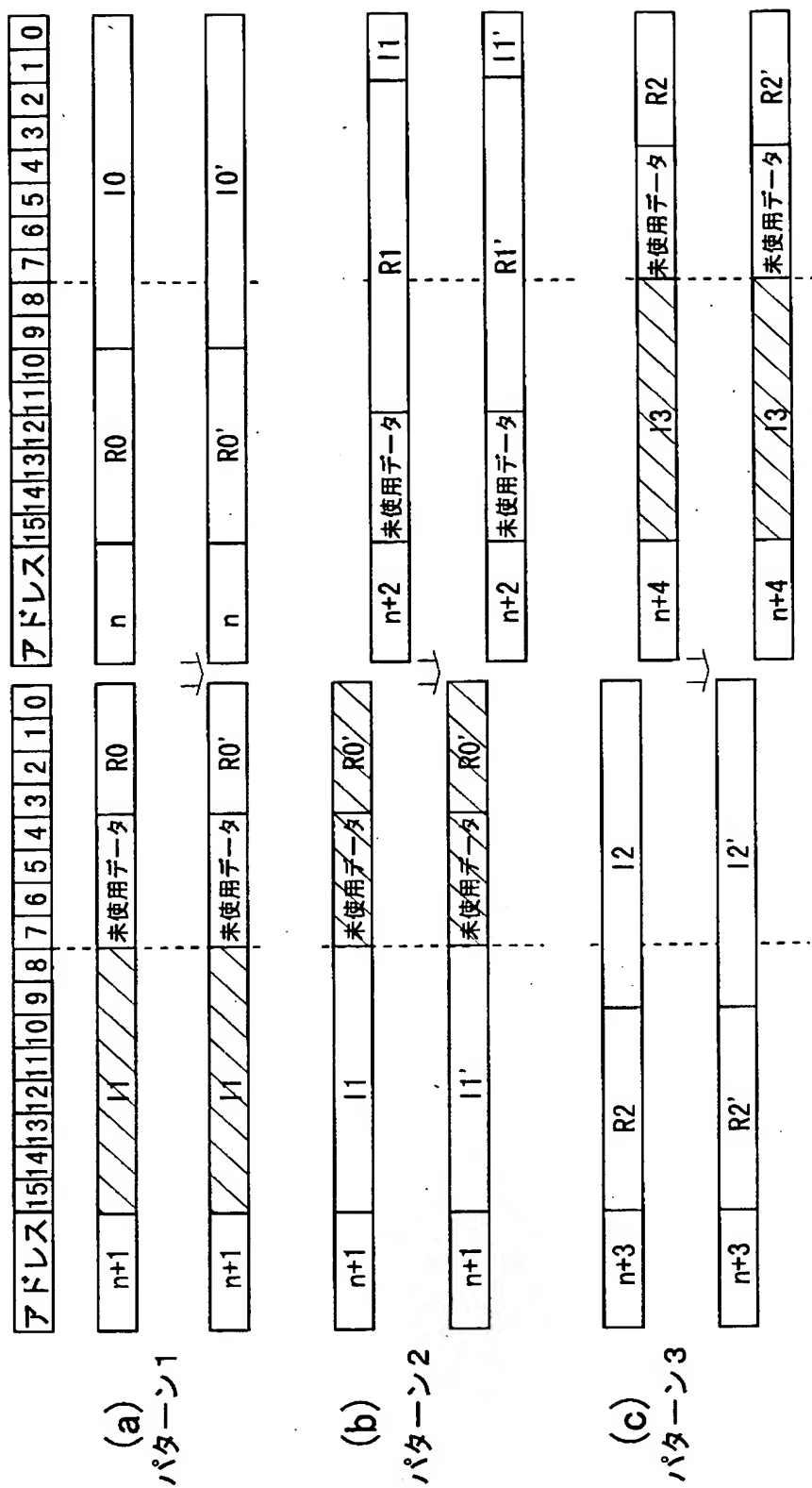
第1のメモリブロックのデータ

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
361	-	-	1	2	4	1	-	-	*	*	*	*	-	-	-	-
363	R	2	4	2	-	-	-	-	-	1	2	4	2	-	-	-
365	*	*	*	*	-	-	-	-	R	2	4	3	-	-	-	-
367	-	-	1	2	4	5	-	-	*	*	*	*	-	-	-	-
369	R	2	4	6	-	-	-	-	-	1	2	4	6	-	-	-
371	*	*	*	*	-	-	-	-	R	2	4	7	-	-	-	-
373	-	-	1	2	4	9	-	-	*	*	*	*	-	-	-	-
375	R	2	5	0	-	-	-	-	-	1	2	5	0	-	-	-
377	*	*	*	*	-	-	-	-	R	2	5	1	-	-	-	-
379	-	-	1	2	5	3	-	-	*	*	*	*	-	-	-	-
381	R	2	5	4	-	-	-	-	-	1	2	5	4	-	-	-
383	*	*	*	*	-	-	-	-	R	2	5	5	-	-	-	-

第2のメモリブロックのデータ

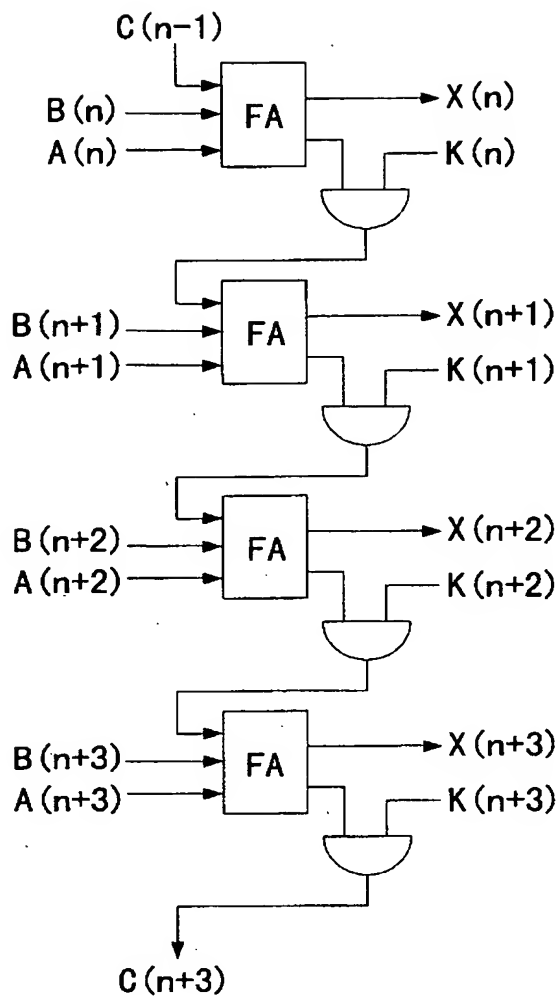
2つのメモリブロックに格納されるデータの配置を示す図

【图 13】



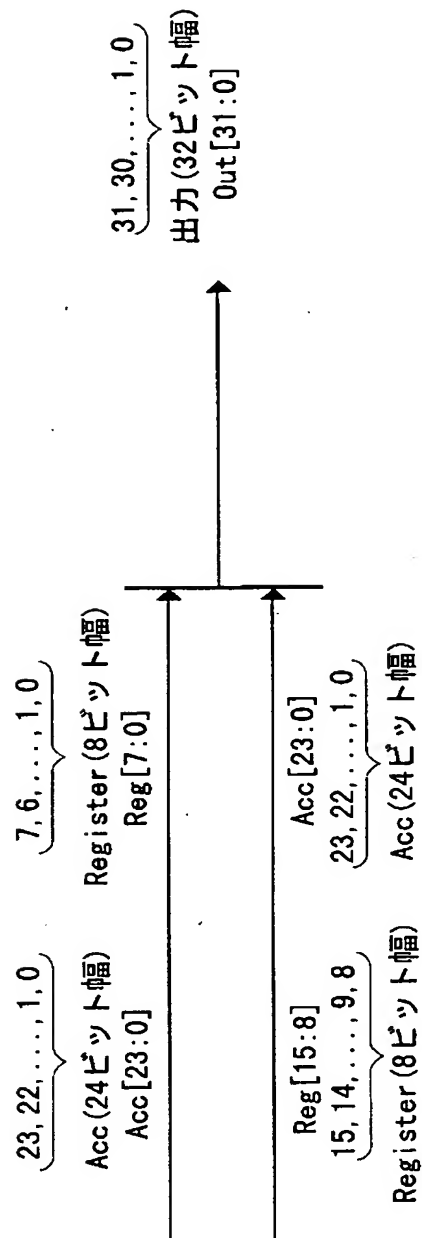
2つのメモリブロックに格納されるデータの遷移過程を示す図

【図 1 4】



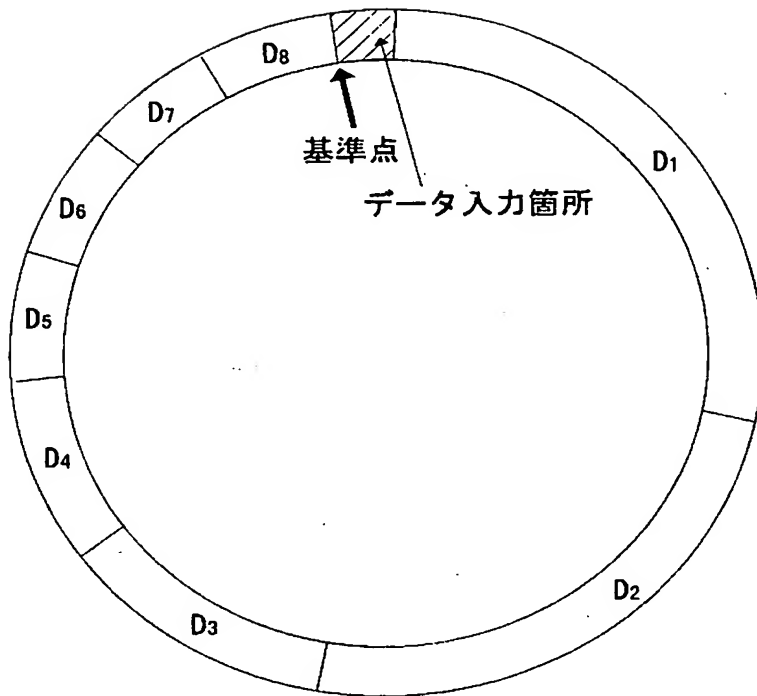
ALUの内部構成を示す図

【図 15】



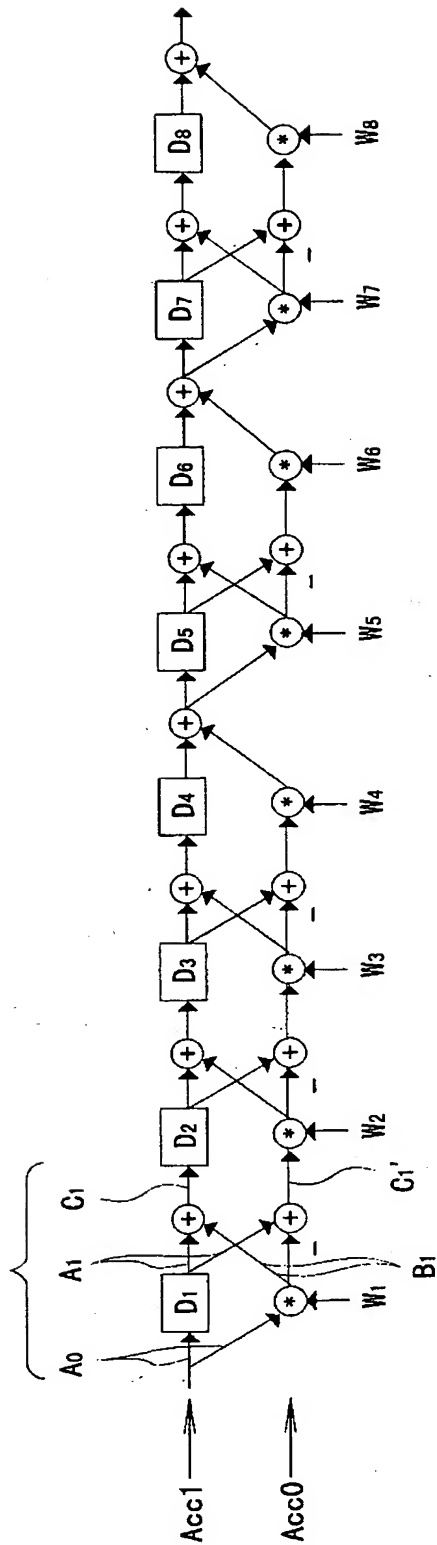
混合回路における出力の遷移を示す図

【図 1 6】



サイクリックなデータの配置を示す図

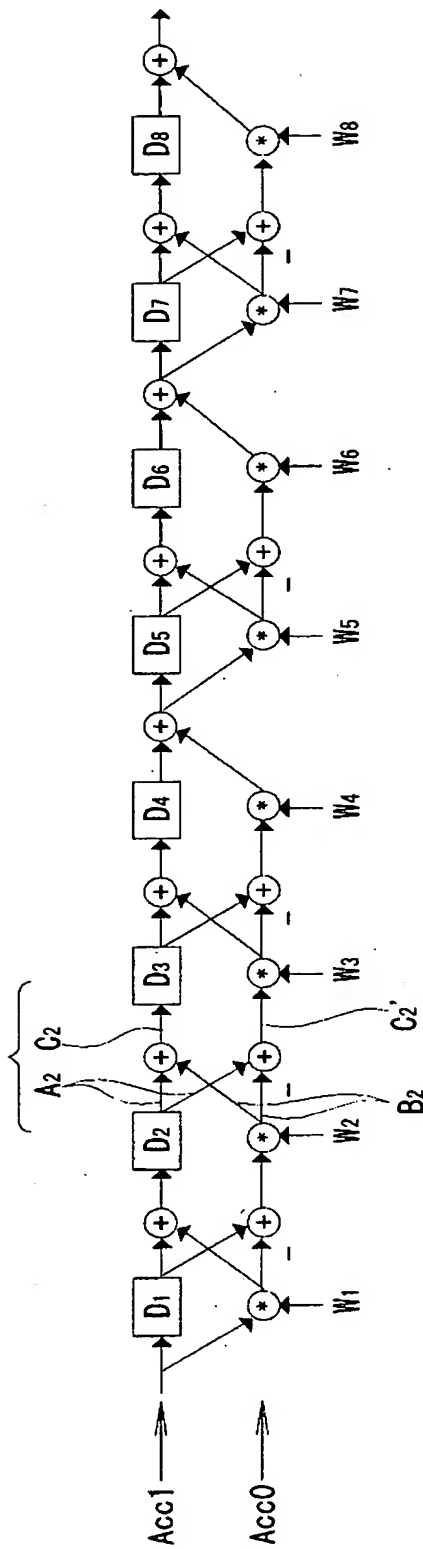
【図 1 7】



$W_{1...8} = \{1, -1, 1, 1, 1, 1, 1, 1\}$
 $D_{1...8} = \{128, 64, 16, 32, 8, 1, 4, 2\}$

DMFアルゴリズムの概要を示す図

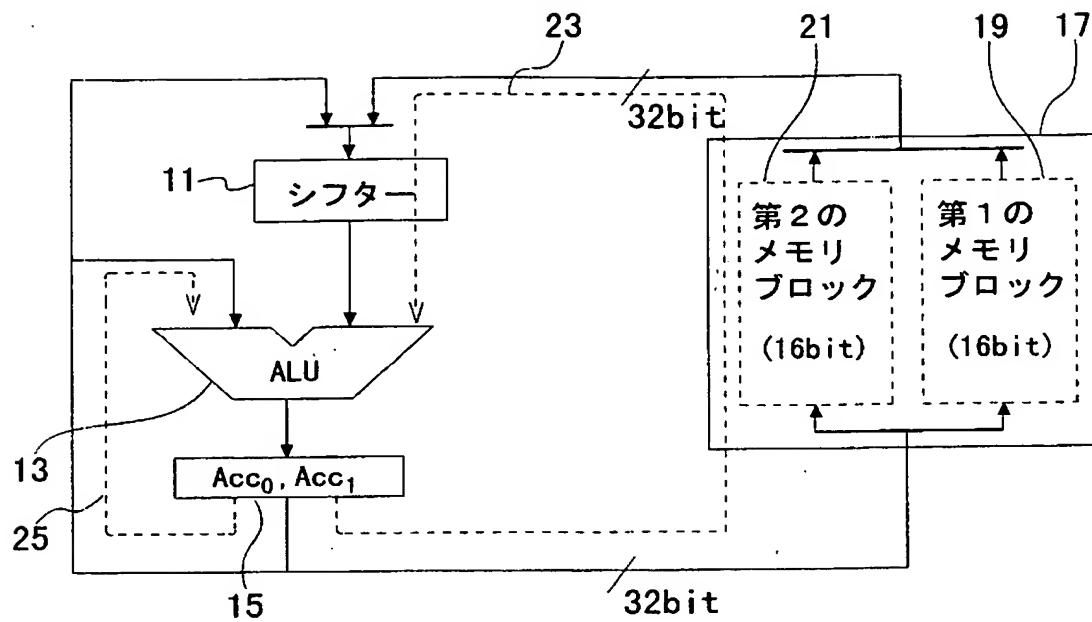
【図 18】



$W_{1...8} = \{1, -1, 1, 1, 1, 1, 1, 1\}$
 $D_{1...8} = \{128, 64, 16, 32, 8, 1, 4, 2\}$

DMFアルゴリズムの概要を示す図

【図19】



従来の演算装置の構成を示す図

【図 2 0】

アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n + 1	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	R0								
n + 3	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	R1								
n + 5	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	R2								
n + 7	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	未使用データ	R3								

第 2 のメモリブロックのデータ

第 1 のメモリブロックのデータ

2 つのメモリブロックに格納されるデータの配置を示す図

【図 21】

アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	*	*	*	*	*	*	*	-	-	-	-	1	0	-	-	-
2	*	*	*	*	*	*	*	-	-	-	-	1	1	-	-	-
4	*	*	*	*	*	*	*	-	-	-	-	1	2	-	-	-
6	*	*	*	*	*	*	*	-	-	-	-	1	3	-	-	-
8	*	*	*	*	*	*	*	-	-	-	-	1	4	-	-	-
10	*	*	*	*	*	*	*	-	-	-	-	1	5	-	-	-
12	*	*	*	*	*	*	*	-	-	-	-	1	6	-	-	-
14	*	*	*	*	*	*	*	-	-	-	-	1	7	-	-	-
16	*	*	*	*	*	*	*	-	-	-	-	1	8	-	-	-
18	*	*	*	*	*	*	*	-	-	-	-	1	9	-	-	-
.
490	*	*	*	*	*	*	*	-	-	-	-	2	4	6	-	-
492	*	*	*	*	*	*	*	-	-	-	-	2	4	7	-	-
494	*	*	*	*	*	*	*	-	-	-	-	2	4	8	-	-
496	*	*	*	*	*	*	*	-	-	-	-	2	4	9	-	-
498	*	*	*	*	*	*	*	-	-	-	-	2	5	0	-	-
500	*	*	*	*	*	*	*	-	-	-	-	2	5	1	-	-
502	*	*	*	*	*	*	*	-	-	-	-	2	5	2	-	-
504	*	*	*	*	*	*	*	-	-	-	-	2	5	3	-	-
506	*	*	*	*	*	*	*	-	-	-	-	2	5	4	-	-
508	*	*	*	*	*	*	*	-	-	-	-	2	5	5	-	-

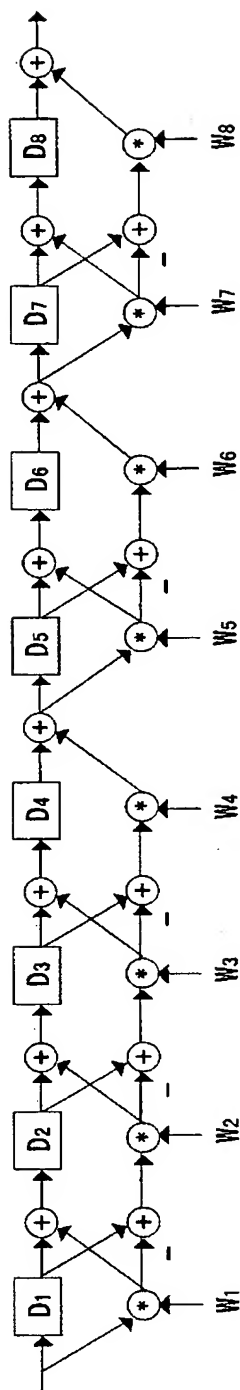
アドレス	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1	*	*	*	*	*	*	*	-	-	-	-	R	0	-	-	-
3	*	*	*	*	*	*	*	-	-	-	-	R	1	-	-	-
5	*	*	*	*	*	*	*	-	-	-	-	R	2	-	-	-
7	*	*	*	*	*	*	*	-	-	-	-	R	3	-	-	-
9	*	*	*	*	*	*	*	-	-	-	-	R	4	-	-	-
11	*	*	*	*	*	*	*	-	-	-	-	R	5	-	-	-
13	*	*	*	*	*	*	*	-	-	-	-	R	6	-	-	-
15	*	*	*	*	*	*	*	-	-	-	-	R	7	-	-	-
17	*	*	*	*	*	*	*	-	-	-	-	R	8	-	-	-
19	*	*	*	*	*	*	*	-	-	-	-	R	9	-	-	-
.
491	*	*	*	*	*	*	*	-	-	-	-	R	2	4	6	-
493	*	*	*	*	*	*	*	-	-	-	-	R	2	4	7	-
495	*	*	*	*	*	*	*	-	-	-	-	R	2	4	8	-
497	*	*	*	*	*	*	*	-	-	-	-	R	2	4	9	-
499	*	*	*	*	*	*	*	-	-	-	-	R	2	5	0	-
501	*	*	*	*	*	*	*	-	-	-	-	R	2	5	1	-
503	*	*	*	*	*	*	*	-	-	-	-	R	2	5	2	-
505	*	*	*	*	*	*	*	-	-	-	-	R	2	5	3	-
507	*	*	*	*	*	*	*	-	-	-	-	R	2	5	4	-
509	*	*	*	*	*	*	*	-	-	-	-	R	2	5	5	-

第1のメモリブロックのデータ

第2のメモリブロックのデータ

2つのメモリブロックに格納されるデータの配置を示す図

【図 2 2】

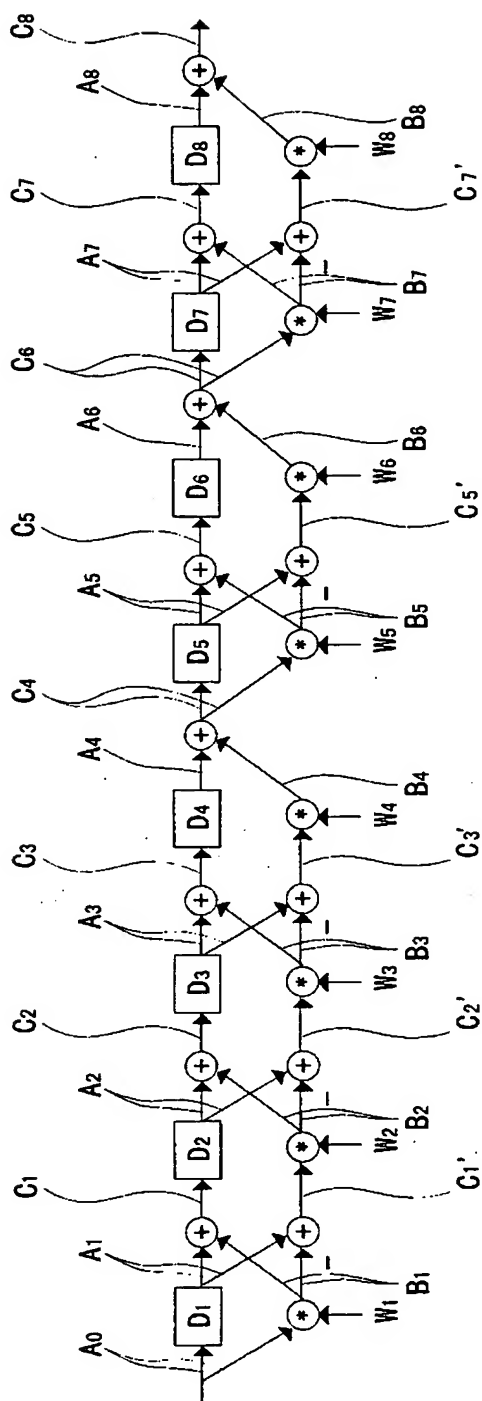


$W_{1...8} = \{1, -1, 1, 1, 1, 1, 1, 1\}$

$D_{1...8} = \{128, 64, 16, 32, 8, 1, 4, 2\}$

DMFアルゴリズムの概要を示す図

【図 23】


$$W_{1\dots 8} = \{1, -1, 1, 1, 1, 1, 1, 1\}$$
$$D_{1\dots 8} = \{128, 64, 16, 32, 8, 1, 4, 2\}$$

DMFアルゴリズムの概要を示す図

【図 2 4】

A	データ	B	算術式	C	算術式	C'	算術式
A ₀	x ₀	—		—		—	
A ₁	x ₁₂₈	B ₁	$W_1 \times A_0$	C ₁	$A_1 + B_1$	C' ₁	$A_1 - B_1$
A ₂	x ₁₉₂	B ₂	$W_2 \times C_1$	C ₂	$A_2 + B_2$	C' ₂	$A_2 - B_2$
A ₃	x ₂₀₈	B ₃	$W_3 \times C_2$	C ₃	$A_3 + B_3$	C' ₃	$A_3 - B_3$
A ₄	x ₂₄₀	B ₄	$W_4 \times C_3$	C ₄	$A_4 + B_4$	—	
A ₅	x ₂₄₈	B ₅	$W_5 \times C_4$	C ₅	$A_5 + B_5$	C' ₅	$A_5 - B_5$
A ₆	x ₂₄₉	B ₆	$W_6 \times C_5$	C ₆	$A_6 + B_6$	—	
A ₇	x ₂₅₃	B ₇	$W_7 \times C_6$	C ₇	$A_7 + B_7$	C' ₇	$A_7 - B_7$
A ₈	x ₂₅₅	B ₈	$W_8 \times C_7$	C ₈	$A_8 + B_8$	—	

DMF アルゴリズムの算術式を示す図表

【図 2 5】

B	算術式	算術式
—		
B ₁	$W_1 \times A_0$	x_0
B ₂	$W_2 \times C_1$	$-(x_{128} - x_0)$
B ₃	$W_3 \times C_2$	$x_{192} + (x_{128} - x_0)$
B ₄	$W_4 \times C_3$	$x_{208} - (x_{192} + (x_{128} - x_0))$
B ₅	$W_5 \times C_4$	$x_{240} + x_{208} - (x_{192} + (x_{128} - x_0))$
B ₆	$W_6 \times C_5$	$x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0)))$
B ₇	$W_7 \times C_6$	$x_{249} + x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0)))$
B ₈	$W_8 \times C_7$	$x_{253} - (x_{249} + x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0))))$

DMF アルゴリズムの算術式を示す図表

【図 26】

C	算術式	算術式
—		
C ₁	A ₁ +B ₁	$x_{128} + x_0$
C ₂	A ₂ +B ₂	$x_{192} - (x_{128} - x_0)$
C ₃	A ₃ +B ₃	$x_{208} + x_{192} + (x_{128} - x_0)$
C ₄	A ₄ +B ₄	$x_{240} + x_{208} - (x_{192} + (x_{128} - x_0))$
C ₅	A ₅ +B ₅	$x_{248} + x_{240} + x_{208} - (x_{192} + (x_{128} - x_0))$
C ₆	A ₆ +B ₆	$x_{249} + x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0)))$
C ₇	A ₇ +B ₇	$x_{253} + x_{249} + x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0)))$
C ₈	A ₈ +B ₈	$x_{255} + x_{253} - (x_{249} + x_{248} - (x_{240} + x_{208} - (x_{192} + (x_{128} - x_0))))$

DMFアルゴリズムの算術式を示す図表

【図 27】

C'	算術式	算術式
—		
C ₁ '	A ₁ -B ₁	$x_{128}-x_0$
C ₂ '	A ₂ -B ₂	$x_{192}+(x_{128}-x_0)$
C ₃ '	A ₃ -B ₃	$x_{208}-(x_{192}+(x_{128}-x_0))$
—		
C ₅ '	A ₅ -B ₅	$x_{248}-(x_{240}+x_{208}-(x_{192}+(x_{128}-x_0)))$
—		
C ₇ '	A ₇ -B ₇	$x_{253}-(x_{249}+x_{248}-(x_{240}+x_{208}-(x_{192}+(x_{128}-x_0))))$
—		

DMFアルゴリズムの算術式を示す図表

【書類名】 要約書

【要約】

【課題】 従来の演算装置は、データをメモリに格納する際に、IパートとRパートの間に未使用データを配置していたため、データの伝送処理や演算処理における効率を低下させていた。

【解決手段】 演算装置は、メモリから読み出したデータに対して所定の演算を行う演算論理ユニット13'と、メモリから読み出したデータを一時格納するレジスタ27と、演算論理ユニットまたはレジスタを選択し、メモリから読み出したデータの一部を、選択した演算論理ユニットまたはレジスタから出力される出力データに置き換えることが可能な混合回路29と、を有することを特徴とする。

【選択図】 図1

認定・付加情報

特許出願の番号	特願2002-241371
受付番号	50201240575
書類名	特許願
担当官	第七担当上席 0096
作成日	平成14年 8月23日

<認定情報・付加情報>

【提出日】	平成14年 8月22日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000000295]

1. 変更年月日 1990年 8月22日
[変更理由] 新規登録
住 所 東京都港区虎ノ門1丁目7番12号
氏 名 沖電気工業株式会社